Πανεπιστήμιο Μακεδονίας

Πρόγραμμα Μεταπτυχιακών Σπουδών

Στην Αναλυτική των Επιχειρήσεων και Επιστήμη των Δεδομένων

Τεχνητή νοημοσύνη και ανοιχτά κυβερνητικά δεδομένα

Διπλωματική Εργασία της

Μαλιόγλου Ζουμπουλιάς Ευαγγελίας

Του Αντωνίου

Θεσσαλονίκη, Μάιος 2022

Ευχαριστίες

Φτάνοντας στο τέλος των ακαδημαϊκών μου χρόνων ήρθε η στιγμή για την εκπόνηση της διπλωματικής μου εργασίας. Δεν θα μπορούσα να μην ευχαριστήσω πρωτίστως τον αδερφό μου Γιώργο, τους γονείς μου αλλά και τη γιαγιά μου για την θερμή τους υποστήριξη όλα αυτά τα χρόνια. Η εμψύχωση, η στήριξη αλλά και η ανοχή που έδειξαν με βοήθησε να συνεχίσω και να ολοκληρώσω τους στόχους και τις σπουδές μου. Σας ευχαριστώ όλους.

Περίληψη

Η παρούσα διπλωματική εργασία εστιάζει στην εφαρμογή της τεχνικής νοημοσύνης σε ανοιχτά κυβερνητικά δεδομένα. Στο πρώτο κεφάλαιο γίνεται μία βιβλιογραφική έρευνα γύρω από τα ανοιχτά κυβερνητικά δεδομένα. Αρχικά δίνεται ο ορισμός των ανοιχτών δεδομένων, και πως ξεκίνησε η χρήση και η αξιοποίηση τους. Στην συνέχεια γίνεται αναφορά στα οφέλη και στα πλεονεκτήματα που μπορούν να έχουν και οι κρατικές υπηρεσίες αλλά και οι πολίτες. Επιπροσθέτως παρουσιάζονται τα απαραίτητα χαρακτηριστικά που θα πρέπει να υπάρχουν ώστε να διασφαλίζεται η ποιότητα και η αξιοπιστία των δεδομένων. Στο δεύτερο μέρος αναφέρεται στο θεωρητικό υπόβαθρο της επιστήμης των δεδομένων. Συγκεκριμένα παρουσιάζονται η μηχανική μάθηση και η βαθιά μηχανική μάθηση. Αναλύεται σε βάθος τα νευρωτικά δίκτυα και συγκεκριμένα ο αλγόριθμός Long Short Term Memory (LSTM). Ακόμη παρουσιάζονται οι τεχνικές των αλγορίθμων δέντρων αποφάσεων, και συγκεκριμένα αναφέρεται ο xgboost. Στο τελευταίο στάδιο γίνεται η ανάπτυξη και ο σχεδιασμός των προβλεπτικών μοντέλων LSTM και xgboost, χρησιμοποιώντας τα ΑΚΔ της Σκωτίας. Βλέπουμε την σύγκριση των δύο προγνωστικών μοντέλων και την τελική τους αξιολόγηση.

Λέξεις Κλειδιά: Open Government Data (OPG), Machine Learning, Deep learning, Neutral Network, Long-Short Term Memory (LSTM), xgboost,

Abstract

The objective of this thesis is the application of artificial intelligent in open government data. Firstly, it is given the definition of open data, and a background history about their use. Then there is a reference in advantages that both governments and citizens could benefit from. In addition, follows the definition of the standard principles of data to ensure data quality and integrity. The second part refers to theoretical background of data science, by presenting machine learning and deep learning. Then analyzing the neural networks, and specifically the Long Short Term Memory algorithm (LSTM). Also, we mention the technics for decision tree algorithm, by focusing in xgboost. In the last chapter we develop two different predictive model, LSTM and xgboost, by using open data of Scotland. Finally, we compare the prediction and performance of the above-mentioned model and we evaluate them.

Key Words: Open Government Data (OPG), Scotland, Machine Learning, Deep learning, Neutral Network, Long-Short Term Memory (LSTM), xgboost,

Πίνακας Περιεχομένων

Κατάλογος Εικόνων

Εισαγωγή

1.1 Περιγραφή του προβλήματος.

Η μηχανική μάθηση υπάρχει εδώ και αρκετό καιρό, αλλά πλέον φαίνεται να είναι πιο επίκαιρη από ποτέ. Την τελευταία πενταετία μάλιστα γίνονται συνεχώς συζητήσεις γύρω από τα ψηφιακά δεδομένα που υπάρχουν, και τη οφέλη μπορεί να δώσει η χρησιμοποίηση τους. Η νέα ψηφιακή κοινωνία είχε επιφέρει σημαντικές αλλαγές στον τρόπο αποθήκευσης, συλλογής και ανάλυσης δεδομένων. Ολοένα και περισσότεροι άνθρωποι συνδέονται καθημερινά στον ψηφιακό κόσμο δημιουργώντας δεδομένα. Υπολογίζεται ότι καθημερινά δημιουργούνται και αποθηκεύονται πάνω από 2,5 εκατομμύριο byte δεδομένων, που φαίνεται μάλιστα να αυξάνεται καθημερινά με γεωμετρική πρόοδο. Ο μεγάλος όγκος δεδομένων σε συνδυασμό με τις νέες τεχνολογίες που επιτρέπουν τη συλλογή και επεξεργασία μεγάλων δεδομένων έχει επιφέρει σημαντικές αλλαγές στον τομέα των πληροφοριακών δεδομένων.

Μέσα σε αυτό πλαίσια αναπτύχθηκε και η επιστήμη των δεδομένων, η οποία παρότι υπάρχει εδώ και λίγες δεκαετίες, τα τελευταία χρόνια χρησιμοποιείται ολοένα και περισσότερο σε ποικίλες εφαρμογές. Ένας σημαντικός τομέας των δεδομένων είναι η μηχανική μάθηση. Σύμφωνα με τον ορισμό ως μηχανικής μάθηση θεωρείται «Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P, αν η απόδοσή του σε εργασίες από το T, όπως μετριέται από το P, βελτιώνεται μέσω της εμπειρίας E. Mitchell (1997).

Με τη χρήση της μηχανικής μάθησης, τα δεδομένα μπορούν να αναλυθούν, και να εξάγουν σημαντικά συμπεράσματα και αποτελέσματα για κάθε πρόβλημα. Όλοι οι μεγάλοι οργανισμοί και επιχειρήσεις έχουν κατανοήσει τη σημασία και τα οφέλη των δεδομένων, και αναπτύσουν συστήματα για την καλύτερη επεξεργασία τους. Πλέον με τη χρήση της μηχανικής μάθησης μπορούν να ληφθούν αποτελεσματικότερες αποφάσεις, οι οποίες θα βασίζονται στα γεγονότα και όχι μόνο στην εμπειρική γνώση (data driven decision). Επίσης

συχνά αναπτύσσονται προβλεπτικά μοντέλα, που επιτυγχάνουν εκπληκτικά ακριβείς προβλέψεις.

Υπάρχουν διάφοροι τομείς και οργανισμοί που μπορούν να έχουν οφέλη από την μηχανική μάθηση και οι κυβερνήσεις είναι ένας από αυτούς. Τα δεδομένα μπορούν να χρησιμοποιηθούν αποτελεσματικά για να βελτιώσουν τις δημόσιες υπηρεσίες αλλά και τις διαδικασίες λήψης αποφάσεων και πολιτικής. (Lyon, 2015) (Martin, Begany, 2017).

Η ολοένα αύξηση των δεδομένων έχει ως συνέπεια της ανάπτυξη των τεχνολογικών συστημάτων, και σε όρους τεχνικής νοημοσύνης αλλά και σε όρους αποθήκευσης δεδομένων μέσω Data Warehouses. Μάλιστα ο τεράστιος όγκος των δεδομένων που υπάρχει αποθηκευμένος, δημιουργήσει τον όρο "Big Data". Σύμφωνα με τον ορισμό, ως μεγάλα δεδομένων θεωρούνται εκείνα τα δεδομένα στα οποία υπάρχει ποικιλία (variety), έχουν συνεχώς αυξανόμενο όγκο (volume), και γρήγορες ταχύτητες (velocity) (Gandomi, Haider, 2014). Τα συγκεκριμένα χαρακτηριστικά μάλιστα είναι γνωστά στη βιβλιογραφία ως τα 3V.

Αν και συχνά συγχέονται τα big data με τα open data στην ουσία πρόκειται για διαφορετικές έννοιες. Ενώ λοιπόν σύμφωνα και με τον ορισμό τους τα μεγάλα δεδομένα επικεντρώνονται κυρίων στην εκμετάλλευση του όγκου τους, τα ανοιχτά δεδομένα δημιουργούν αξία μέσω της διάθεσή τους προς τρίτους. Τα ανοιχτά δεδομένα πρέπει να είναι ελεύθερα προς χρήση στο διαδίκτυο.

Πολλές χώρες και οργανισμοί έχουν αρχίσει να χρησιμοποιούν τα δεδομένα με την χρήση τεχνολογιών. Είναι γεγονός ότι έχουν ήδη αναγνωριστεί τα οφέλη της μηχανικής μάθησης και αυτός είναι ο βασικός λόγος που παρατηρείται η τάση των κυβερνήσεων να εστιάζουν και να επικεντρώνονται στην κατασκευή αλγορίθμων. Ωστόσο η κατασκευή ενός δίκαιου και αντικειμενικού αλγορίθμου δεν είναι εύκολο να επιτευχθεί. Το πιο σημαντικό βήμα για τον σχεδιασμό ενός αντικειμενικού αλγορίθμου είναι να εξασφαλίσουν οι κυβερνήσεις ποιοτικά και αμερόληπτα δεδομένα. Ο ΟΟΣΑ δηλώνει ότι μία από τις βασικές προκλήσεις της μηχανικής μάθησης είναι να διασφαλίσει ότι τα συστήματα της είναι αξιόπιστα και ότι καλύπτονται οι εθνικές πολιτικές. (OECD, 2019). Μάλιστα αρκετές αρκετοί υποστηρίζουν

ότι οι περισσότερες κυβερνήσεις απλά δεν είναι έτοιμες ακόμη για την μηχανική μάθηση και θα πρέπει πρώτα να επικεντρωθούν στην συλλογή των δεδομένων τους.

Υπάρχουν διάφοροι τρόποι με τους οποίους μία κυβέρνηση μπορεί να συλλέξει τα δεδομένα και τις πληροφορίες των πολιτών αλλά και τον οργανισμών που υπάρχουν σε αυτήν. Συχνά αυτό συμβαίνει με ερωτηματολόγια, συνεντεύξεις και πειράματα. Ωστόσο για να γίνει επιτυχώς η συλλογή των στοιχείων, απαραίτητη προϋπόθεση είναι βέβαια η κυβέρνηση να χτίσει μία σχέση εμπιστοσύνης με τους πολίτες της. Σε διαφορετική περίπτωση, έχει παρατηρηθεί πως τα ίδια τα άτομα δεν είναι πρόθυμα να μοιράσουν τα προσωπικά τους δεδομένα με την κυβέρνηση ακόμη και αν αυτό είναι προς όφελος τους. Επίσης αν η συλλογή γίνεται όχι με ελεύθερη βούληση, αλλά επιβάλλεται από τους κυβερνώντες, τότε υπάρχει το ενδεχόμενα να συμβεί διαστρέβλωση των δεδομένων, μέσω μη αληθών καταχωρίσεων.

Για να μπορέσουν τα άτομα να είναι σε θέση να διαμοιρασθούν πληροφορίες και προσωπικά δεδομένα προς την κυβέρνηση είναι πολύ σημαντικό τα συστήματα τεχνητής νοημοσύνης αλλά και η αποθήκευσης των πληροφοριών να εγγυώνται στους πολίτες προστασία της ιδιωτικής ζωής. Επίσης θα πρέπει όλα τα προσωπικά δεδομένα και οι πληροφορίες να προστατεύονται από διαδικτυακές επιθέσεις ή αλλοίωση των δεδομένων. Αν οι πολίτες αισθανθούν ότι δεν είναι ασφαλές τα δεδομένα τους δεν θα μπορέσει ποτέ η κυβέρνηση να συλλέγει αληθή και αντικειμενικά στοιχεία.

1.2 Αντικείμενο Έρευνας

Αντικείμενο της παρούσας διπλωματικής είναι να διερευνηθεί πως μπορεί να χρησιμοποιηθεί η μηχανική μάθηση χρησιμοποιώντας τα ανοιχτά κυβερνητικά δεδομένα. Με τη χρήσης της μηχανικής μάθησης και συγκεκριμένη της βαθιάς μηχανικής μάθησης σχεδιάστηκε ένα μοντέλο πρόβλεψης τροφοδοτώντας το με ανοιχτά δεδομένα.

Πιο συγκεκριμένα μελετήθηκε η περίπτωση της Σκωτίας, μιας χώρας με ΑΚΒ και εφαρμόστηκε μοντέλο τεχνητής νοημοσύνης για την πρόβλεψης της κίνησης των τιμών των σπιτιών σε συγκεκριμένες γεωγραφικές περιοχές (2011 Data Zones). Σκοπός της εργασίας

ήταν να διασαφηνιστεί ο τρόπος με τον οποίον η χρήση της μηχανικής μάθησης στα ανοιχτά δεδομένα θα μπορούσε να επηρεάσει την λήψη των αποφάσεων στη δημοσιονομική πολιτική της χώρας.

Στην παρούσα εργασία γίνεται ανάλυση των νευρωτικών δικτύων, μοντέλων δηλαδή που ανήκουν στον τομέα της βαθιάς μηχανικής μάθησης (deep learning). Επίσης αναφέρεται και το του μοντέλου xgboost, ένας από τους πιο δημοφιλής αλγορίθμους στο kaggle. Έπειτα γίνεται η σύγκριση των δύο μοντέλων και των αποτελεσμάτων τους. Ακόμη εξετάζει η συμβολή τους στον τρόπο επίλυσης του ζητήματος και κατά πόσο μπόρεσαν να προσφέρουν ακριβείς προβλέψεις. Τέλος γίνεται εφαρμογή των αλγορίθμων με γλώσσα προγραμματισμού Python και εξετάζονται βιβλιοθήκες όπως το keras.

2. Ανοιχτά κυβερνητικά Δεδομένα

2.1 Τι είναι τα ανοιχτά κυβερνητικά δεδομένα.

Την τελευταία δεκαετία έχουν έρθει στο προσκήνιο αρκετές συζητήσεις των κρατών σχετικά με τη χρήση των ανοιχτών δεδομένων στην κρατική διακυβέρνηση. Ήδη πολλές χώρες έχουν κατανοήσει τη σημασία και τα οφέλη των ΑΚΔ, αφού πλέον έχει γίνει το επίκεντρο της παγκόσμιας προσοχής, και έχουν αναπτύξει πολιτικές Ανοικτής Κυβέρνησης Δεδομένων.

Μέσα σε αυτό το πλαίσιο έχει πλέον αναπτυχθεί ο όρος Government 2.0 οποίος αναφέρεται για να περιγράψει τις κυβερνήσεις εκείνες που αναπτύσσουν πολιτικές για την αξιοποίηση της τεχνολογίας με τη χρήση δεδομένων και εργαλείων ανοιχτού κώδικα. Πολλοί οργανισμοί και αντίστοιχες πρωτοβουλίες έχουν δημιουργηθεί τα τελευταία χρόνια με σκοπό την ανάπτυξη των ΑΚΔ.

Το Ηνωμένο βασίλειο που αναλύεται στην παρούσα εργασία ανήκει μάλιστα σε μία από τις οκτώ ιδρύτριες χώρες του Open government partnership (OGP), μαζί με τη Βραζιλία, την Ινδονησία, το Μεξικό, τη Νορβηγία, τις Φιλιππίνες, τη Νότια Αφρική, και Ηνωμένες Πολιτείες. Ο συγκεκριμένος οργανισμός ιδρύθηκε το 2011 έχοντας ως κύριο στόχο την

προώθηση της ανοιχτής διακυβέρνησης. Πλέον στα μέλη του ανήκουν 77 χώρες, οι οποίες μοιράζονται το όραμα για ένα κράτος που θα αξιοποιεί τα δεδομένα με την χρήση νέων τεχνολογιών προς όφελος των πολιτών.

Με την έννοια των ανοιχτών δεδομένων θεωρούνται όλα τα δεδομένα που είναι προσβάσιμα σε όλους (μέσα από το διαδίκτυο), τα οποία δημοσιεύονται χωρίς τεχνικούς περιορισμούς, αλλά διέπονται από ειδική άδεια όπου επιτρέπει την επαναχρησιμοποίηση τους. (Smith et. al, 2008). Με τον όρο ανοιχτά δεδομένα εννοούνται δηλαδή οποιαδήποτε δεδομένα συνάδουν με τον παραπάνω ορισμό. Αυτά μπορεί να είναι Μητρώα Πολιτών, εκδόσεις διαβατηρίων και αδειών, οικονομικές πληροφορίες των πολιτών, γεωγραφικές πληροφορίες (χάρτες, οδικά δίκτυα, κ.ά), νομικές πληροφορίες, αλλά και κοινωνικές πληροφορίες που διέπουν το κράτος, όπως στατιστικά στοιχεία της κοινωνίας, δείκτες ανεργίας ανεργίας, πληροφορίες της οικονομίας, αλλά και της υγείας. Ωστόσο αξίζει να σημειωθεί ότι με στην σφαίρα των ανοιχτών δεδομένων δεν ανήκουν τα πολύ προσωπικά και εμπιστευτικά δεδομένα του καθενός. Το γεγονός λοιπόν ότι τα ανοιχτά δεδομένα μπορούν να διαμοιραστούν και να επαναχρησιμοποιηθούν ελεύθερα από όλους συντελεί στην ανάπτυξη ενός κλίματος συνεργασίας αλλά και ανάπτυξης της δημιουργικότητας στο γενικό πλαίσιο (Hofmokl, 2010).

Ωστόσο η προσέγγιση μιας κυβέρνησης σχετικά με την υιοθέτηση ή όχι ΑΚΔ δεν έχει μόνο δύο πλευρές, αλλά αντιθέτως θα πρέπει να αντιμετωπίζεται ως ένα πολυδιάστατο σύνθετο έργο. Οι κυβερνήσεις θα πρέπει να λάβουν υπόψη την διαφορετική κουλτούρα κάθε κράτους και λαού αλλά και την ίδια την διαφορετικότητα των ανθρώπων κυρίως όσον αφορά στον τρόπο πρόσβασης και χρήσης των δημόσιων δεδομένων. Έτσι ως ένα πολυσύνθετο πρόβλημα τα δεδομένα δεν θα έπρεπε να κατηγοριοποιούνται αυστηρά και απόλυτα είτε ως ανοιχτά ή κλειστά. Αντιθέτως μπορούσαν να χαρακτηριστούν περισσότερο ή λιγότερο ανοιχτά (Smith et. al, 2008).

2.2 Οφέλη ανοικτών δεδομένων

Τα οφέλη από την χρήση των ανοιχτών δεδομένων είναι πλέον διακριτά όχι μόνο από τους κρατικούς μηχανισμούς, αλλά έχουν αρχίσει να γίνονται αντιληπτά και στο ευρύ κοινό. Είναι πλέον κοινώς αποδεκτό ότι με τα ΑΚΔ οι κυβερνήσεις μπορούν να επιτύχουν τη σύνδεση των δημοσίων υπηρεσιών βελτιώνοντας τον τρόπο λειτουργίας των διαδικασιών που απαιτούνται, προσδίδοντας άμεσο όφελος στους πολίτες (Mutuku & Colaco, 2014). Επιπροσθέτως έχει αποδειχθεί ότι με την χρήση των ΑΚΔ τα κράτη μπορούν να επιτύχουν καλύτερους δείκτες διακυβέρνησης, όπως για παράδειγμα ο έλεγχος της διαφθοράς και η λογοδοσία (Rachel Gong & Hui San Chiam, 2019). Ακόμη η χρήση των ΑΚΔ σε συνδυασμό με την μηχανική μάθηση μπορεί να δώσει αποτελεσματικότερη εικόνα για ένα γεγονός και να ληφθούν αποφάσεις βασισμένες στα δεδομένα (data driven decision). Με την ηλεκτρονική, δημόσια δημοσίευση των δεδομένων οι ίδιοι οι πολίτες έχουν πλέον την δυνατότητα συμμετοχής και δράσης στις αποφάσεις του κράτους, ενισχύοντας έτσι την διαφάνεια του κράτους, και καθιστώντας το πιο δημοκρατικό. Είναι γεγονός ότι μία κρατική στρατηγική βασιζόμενη στη χρησιμοποίηση των ανοιχτών κυβερνητικών δεδομένων ενισχύει την καινοτομία, αλλά και την τεκμηριωμένη διαδικασία λήψης αποφάσεων.

2.2.1 Κρατικές υπηρεσίες

Το παραδοσιακό μοντέλο αποθήκευσης των πληροφορίων ενός κράτους ήταν να αποθηκεύονται τα στοιχεία σε έγγραφα, αρχεία και φακέλους. Ο όγκος αυτών των δεδομένων ήταν αρκετά μεγάλος, αλλά το συγκεκριμένο είδος αποθήκευσης δεν διευκόλυνε την αναζήτηση και την ανάκτησή τους. Μάλιστα συχνά τα δεδομένα αυτά βρίσκονται σε διαφορετικές κρατικές ή μη υπηρεσίες και τμήματα. Ως εκ τούτου η συλλογή, η συγχώνευση αλλά και ο συνδυασμός πληροφοριών από διαφορετικές πηγές προκαλούσαν πάντα μία σύγχυση, και ενίσχυε το πρόβλημα της κρατικής γραφειοκρατίας. Ακόμη λόγω των ανωτέρω δυσκολιών, συχνά υπάρχει ελλιπής πληροφόρηση αναφορικά με συγκεκριμένες διαδικασίες, και έτσι προκύπταν διεκπερωτικά λάθη αλλά και ανακρίβειες. Ειδικά στις περιπτώσεις των χειρόγραφων εγγραφών ήταν συχνό φαινόμενα οι αναγραμματισμοί των στοιχείων, με αποτέλεσμα να χολώνεται η σύγκριση των δεδομένων διαφορετικών πηγών. Επίσης δεν είναι εύκολη και η διόρθωση των στοιχείων, καθώς συχνά θα πρέπει η ίδια διόρθωση να γίνει σε πολλά διαφορετικά έγγραφα.

Με την χρήση όμως ενός πλαισίου ανοιχτών δεδομένων και μιας ενιαίας στρατηγικής διαχείρισης των δεδομένων, θα μπορούσε να αλλάξει ο κυβερνητικός μηχανισμός. Οι διαδικασίες θα μπορούσαν να γίνουν ξεκάθαρες και ευκολονόητες, και τα στοιχεία που απαιτούνται από τους πολίτες θα είναι πλέον εύκολα προσβάσιμα. Με αυτόν τον τρόπο θα επιταχυνόταν οι διαδικασίες ενώ ταυτόχρονα να μειωνόταν το ποσοστό λάθους. Ένα αντίστοιχο παράδειγμα είναι το τεχνολογικό σύστημα που ανέπτυξε η κυβέρνηση του Καναδά στον τομέα της μεταναστευτικής πολιτικής. Πιο συγκεκριμένα αναπτύχθηκε ένα αυτοματοποιημένο σύστημα λήψης απόφασης σχετικά με τον έλεγχο των αιτήσεων προσωρινής βίζας παραμονής (LuciaNalbandian, 2022).

2.2.2 Πολίτες

Με την θέσπιση στρατηγικών βασιζόμενες σε ανοιχτά κυβερνητικά δεδομένα οι κυβερνήσεις επωμίζονται άμεσα τα οφέλη από τα πλεονεκτήματα που συνεπάγονται. Πέραν τούτου υπάρχουν και έμμεσα οφέλη για τις κυβερνήσεις. Το βασικότερο είναι η διαφάνεια που κερδίζει το κράτος. Έχοντας ανοιχτά τα δεδομένα οι ίδιοι οι πολίτες μπορούν να έχουν πλέον ξεκάθαρη εικόνα για κάθε κατάσταση της κυβέρνησης. Ως αποτέλεσμα αυξάνεται η εμπιστοσύνη των πολιτών απέναντι στις κυβερνητικές αποφάσεις και τους κυβερνώντας αφού η διαφάνεια καθιστά πιο το κράτος πιο δημοκρατικό.

Είναι μεγάλης σημασίας το γεγονός ότι οι πολίτες θα μπορούν να αξιολογήσουν την απόδοση των κυβερνητικών αποφάσεων αλλά και να ελέγχουν τους δημόσιους πόρους. Αυτό πρακτικά σημαίνει ότι αφού πλέον οι πολίτες θα έχουν γνώση των δεδομένων της κυβέρνησης, θα πρέπει η κυβέρνηση να είναι πρακτικά έτοιμη να λογοδοτήσει απέναντι στους πολίτες της. Η έννοια της λογοδοσίας της κυβέρνησης ενισχύει φυσικά το δημοκρατικό χαρακτήρα που διέπει τα κράτη. Όπως αναφέρεται και από τον Mark Bovens, στη δημοκρατία θα πρέπει οι έχοντες εξουσία να είναι σε θέση να λογοδοτήσουν δημόσια για τις πράξεις τους, τις αποφάσεις τους, τις πολιτικές τους, καθώς ακόμη και για τις δαπάνες τους. (Bovens 2005, 182).

Μέσα σε ένα κράτος διαφάνειας δημιουργούνται ίσοι όροι ανταγωνισμού μεταξύ πολιτών και κυβέρνησης (Richter, Georgiadou, 2013). Έτσι και οι κυβερνώντες θα γνωρίζουν ότι οι πράξεις είναι γνωστές στο κοινό και κατ' επέκταση κρίνονται από αυτές, αλλά και οι ίδιοι οι πολίτες θα έχουν πιο ανεπτυγμένο το αίσθημα του "ανήκειν" και θα αυξηθεί η αλληλεπίδραση τους με το κράτος. Άλλωστε σε όλες τις κοινωνίες οι πολίτες διεκδικούσαν το δικαίωμα της πληροφόρησης.

Πλέον με την έλευση του διαδικτύου, άνθισε ακόμη περισσότερο η εύκολη και γρήγορη πληροφόρηση, και το άτομο μπορεί να ενημερωθεί άμεσα για διάφορα θέματα. Έτσι είναι λογικό τα άτομα να διεκδικούν πλέον εντονότερα τη γνώση των πληροφοριών και των στοιχείων των κυβερνήσεων. Σε ένα κράτος πλήρως ανοιχτών δεδομένων και διαφάνειας, οι πολίτες θα γνωρίζουν ότι θα μπορούν ακόμη και οι ίδιοι να είναι σε θέση να ελέγξουν τα έξοδα και τις δαπάνες τις κυβερνήσεις και να διαπιστώσουν αν ανταποκρίνονται τα φερόμενα δαπανηθέντα ποσά στις πραγματικές δαπάνες που πραγματοποιήθηκαν για κάποιο δημόσιο έργο.

2.2.3 Επιστημονική Έρευνα

Τέλος τα ανοιχτά δεδομένα παρέχουν άμεσα οφέλη και στην επιστημονική κοινότητα. Είναι γεγονός ότι ο περισσότερος χρόνος μιας επιστημονικής έρευνας απαρτίζεται από τη συλλογή και τη διαλογή αξιόπιστων δομημένων δεδομένων. Με την ύπαρξη λοιπόν των ανοικτών δεδομένων οι επιστήμονες μπορούν γρήγορα και εύκολα να έχουν πρόσβαση στα απαραίτητα στοιχεία που χρειάζονται για τις έρευνές τους. Ακόμη μπορούν να δημιουργηθούν ευκολότερα συνεργασίες, είτε άμεσες, είτε απλώς επαναχρησιμοποιώντας τα δεδομένα από προηγούμενες επιστημονικές έρευνες, παράγοντας έτσι γρηγορότερα επιστημονικό έργο (Katharina Sielemann, Alenka Hafner, Boas Pucker, 2020).

Μάλιστα σύμφωνα με την UNESCO (United Nations Educational, Scientific and Cultural Organization) ως ανοιχτή επιστήμη ορίζεται το κίνημα του να κάνεις επιστημονική έρευνα χρησιμοποιώντας ανοιχτά δεδομένα. Είναι γεγονός ότι για τις χώρες που έχουν υιοθετήσει τα ΑΚΔ σε υψηλό ποσοστό, έχουν πραγματοποιηθεί περισσότερες επιστημονικές έρευνες

(Barbara Ubaldi, 2013). Ένα αντίστοιχο παράδειγμα αποτέλεσαν οι πολλές έρευνες που δημοσιεύθηκαν σχετικά με τον covid-19. Φυσικά είναι ένα επίκαιρο ζήτημα, αλλά αν δεν υπήρχαν πηγές συλλογής ανοιχτών δεδομένων δεν θα διευκολυνόταν ιδιαίτερα η έρευνα.

2.3 Αξιοπιστία Δεδομένων

2.3.1 Ποιότητα δεδομένων και μηχανική μάθηση

Προκειμένου να επωμιστούν τα οφέλη των ΑΚΔ παρατηρείται ότι ολοένα και περισσότερες κυβερνήσεις στρέφονται προς αυτήν την κατεύθυνση. Ωστόσο η δημοσίευση και μόνο των δεδομένων δεν επιφέρει αυτόματα τα ανωτέρω πλεονεκτήματα αλλά χρειάζεται να γίνει εμπεριστατωμένη ανάλυση με επιστημονική έρευνα. Σε πολλές χώρες μάλιστα η κυβέρνηση συνεργάζεται με οργανισμούς και φορείς οι οποίοι επεξεργάζονται και αναλύουν τα κυβερνητικά δεδομένα δημιουργώντας χρήσιμη και αξιολόγηση πληροφορία. Η ανάλυση των δεδομένων γίνεται με την χρήση της μηχανικής μάθησης, και τη δημιουργία κατάλληλων αλγορίθμων και μοντέλων

Φυσικά ανάλογα με το αποτέλεσμα που θέλει να πετύχει το κάθε κράτος χρειάζεται και διαφορετική αξιοποίηση και δράση. Αν μία χώρα θέλει να δημιουργήσει ένα εργαλείο - μηχανισμό για κάποια υπηρεσία της, το οποίο να βασίζεται στα ανοιχτά δεδομένα, απαιτείται ταυτόχρονα υπολογιστική ισχύ.

Βέβαια για μία επιτυχημένη στρατηγική ανοιχτών δεδομένων σημαντικό ρόλο διαδραματίζει η ποιότητα των δεδομένων. Τα δεδομένα υψηλής ποιότητα είναι θεμελιώδης παράγοντας επιτυχημένης στρατηγικής (Haug, Zachariassen, Liempd, 2010). Τα δεδομένα που θα χρησιμοποιηθούν για την άσκηση πολιτικής θα πρέπει να διέπονται από αντικειμενικότητα, να αντικατοπτρίζουν το γενικό σύνολο και μην υπάρχει τάση προκατάληψης έναντι συγκεκριμένων πληθυσμιακών ομάδων. Είναι σημαντικό λοιπόν να ελέγχονται τα δεδομένα που συλλέγονται από τις κυβερνήσεις ώστε να διασαφηνισθεί αν επηρεάζονται έναντι κοινωνικών ομάδων και οι αποφάσεις να έχουν ως κύριο γνώμονα το όφελος στο σύνολο της κοινωνίας.

Έτσι παρατηρείται μία φαινομενικά αυξητική τάση στην ενασχόληση των πολιτών με τα δημόσια δεδομένα. Μάλιστα πολλές φορές οι ίδιοι οι πολίτες δημιουργούν καινοτόμες ιδέες κατά τις οποίες απαιτείται η συλλογή και επεξεργασία των ανοιχτών δεδομένων. Άλλες φορές πάλι αυτό είναι παρακινούμενο μέσα από τους κρατικούς μηχανισμούς. Φυσικά στην περίπτωση που η παρακίνηση γίνεται μέσω του ίδιου του κράτους, η συλλογή αλλά και η αξιοπιστία των δεδομένων εξαρτάται για ακόμη μία φορά από την εμπιστοσύνη που έχουν οι πολίτες προς τους κυβερνώντες. Αρκετές έρευνες έχουν δείξει ότι πιο πιθανόν να μοιραστούν οι πολίτες τα δεδομένα τους, όταν η παρακίνηση γίνεται από ιδιωτικά μέσα ή και από τους ίδιους τους πολίτες παρά όταν υποστηρίζεται από το ίδιο το κράτος. Εδώ βέβαια έρχεται η ανάγκη για ανάπτυξη μιας σχέσης εμπιστοσύνης μεταξύ των χωρών και των πολιτών τους. Για να διαμοιράσουν οι πολίτες τα δεδομένα τους θα πρέπει σίγουρα να έχουν κατανοήσει το σκοπός αυτού αλλά και τα οφέλη που μπορούν να αποκομίσουν από την χρήση τους. Ακόμη θα πρέπει σίγουρα να έχουν εμπιστοσύνη στην κυβέρνηση αλλά και νιώθουν ότι τα δεδομένα τους είναι ασφαλείς.

2.3.2 Επιλογή δείγματος

Η τρόπος συλλογής δεδομένων είναι καίριας σημασίας, αν αναλογιστεί κανείς ότι τα τελικά δεδομένα πρέπει να είναι αμερόληπτα (bias) και καλής ποιότητας. Τα αμερόληπτα δεδομένα έπονται στο γεγονός ότι δεν υπάρχει διάκριση ως προς ορισμένες κοινωνικές ομάδες. Οι κυβερνήσεις θα πρέπει να προετοιμάσουν κατάλληλα εργαλεία ώστε να ελέγχουν την ποιότητα των δεδομένων τους για μεροληψία. Κατά τη διαδικασία ανάλυσης των δεδομένων θα πρέπει να ελεγχθούν τα δεδομένα για τον εντοπισμό διακρίσεων σε συγκεκριμένες ομάδες, όπως για παράδειγμα φυλετικές διακρίσεις, θρησκευτικές, σεξουαλικές ή και ακόμη βάσει συγκεκριμένων εθνικοτήτων. Σαφώς μπορεί να εμφανιστεί μία συσχέτιση μεταξύ κάποιων συγκεκριμένων χαρακτηριστικών, που ωστόσο ίσως πλέον να έχει εξαλειφθεί και απλώς να υπάρχει βάσει των παλαιότερων στοιχείων. Οπότε μία αντίστοιχη ιστορική συσχέτιση δεν αποδεικνύει την κατ’ εξακολούθηση σχέση αιτίου αιτιατού (Barryhill, Heang, Clogher, McBride, 2019).

Τέτοιες ανισότητες είναι εύκολο να δημιουργηθούν και να παραποιήσουν το αποτέλεσμα. Για παράδειγμα κατά το δειγματοληπτικό έλεγχο είναι πολύ σημαντικό να επιλεγεί το ποσοστό του δείγματος που θα είναι αντιπροσωπευτικό του συνόλου του πληθυσμού. Αν επιλεγεί δείγμα βάσει συγκεκριμένης κοινωνίας ομάδας, γεωγραφικής περιοχής, ή κάποιου άλλου διαχωρισμού, και σύμφωνα με αυτό το δείγμα ερμηνευθεί η γενικευμένη απόφαση για το σύνολο του πληθυσμού είναι σχεδόν σίγουρο ότι τα δεδομένα του δείγματος δεν θα είναι αμερόληπτα (bias).

Επίσης υπάρχουν και περιπτώσεις που ενώ επιλέγεται το σύνολο των δεδομένων προς ανάλυση, τα χαρακτηριστικά που προκύπτουν από αυτό το σύνολο δεν συνάδουν με τα χαρακτηριστικά του γενικού συνόλου του πληθυσμού. Το φαινόμενο αυτό εντοπίζεται κυρίως στις αναπτυσσόμενες χώρες. Ο λόγος είναι ότι για να συλλεχθούν διαδικτυακές πληροφορίες των πολιτών, χρειάζεται συχνά να συνδεθούν οι πολίτες στο διαδίκτυο. Αυτό αμέσως προϋποθέτει την ύπαρξη ενός έξυπνου τηλεφώνου "smartphone" και σύνδεση στο διαδίκτυο. Ωστόσο σε αρκετές χώρες χαμηλού εισοδήματος μόνο μία συγκεκριμένη κοινωνική ομάδα του πληθυσμού έχει στην κατοχή της smartphone όπου μπορεί να συνδεθεί ηλεκτρονικά. Συνήθως αυτοί οι χρήστες ανήκουν σε πλουσιότερες τάξεις, με συχνά υψηλό μορφωτικό επίπεδο. Έτσι αν μία αναπτυσσόμενη χώρα προχωρήσει σε λήψη αποφάσεων οι οποίες έχουν επέλθει με ανάλυση δεδομένων των πληροφοριών που "αφήνει" στο διαδίκτυο η συγκεκριμένη κοινωνική τάξη, η οποία προφανώς δεν αντιπροσωπεύει το γενικό σύνολο του πληθυσμού, τότε πιθανών θα έχουν προκύψει λανθασμένα συμπεράσματα (Fung, Gilman, Shkabatur, 2015). Σύμφωνα με το (ITU 2011) μόνο το 20% των συνδρομητής κινητής τηλεφωνίας του αναπτυσσόμενου κόσμου έχουν πρόσβαση στο διαδίκτυο. Δεν υπάρχει λοιπόν η δυνατότητα να ληφθούν αποφάσεις για την ενίσχυση για παράδειγμα της οικονομίας ή της ευημερίας για μία κοινωνική ομάδα που δεν "υπάρχει" στον εικονικό κόσμο των δεδομένων. Έτσι για να είναι δίκαιη και με ισότητα η πολιτική ενός κράτους που βασίζεται στα ΑΚΔ θα πρέπει να διασφαλιστεί η συλλογή των δεδομένων να γίνεται χωρίς αποκλεισμούς.

2.3.3 Νομικό Καθεστώς

Σημαντικό ρόλο ωστόσο για μία επιτυχημένη στρατηγική ανοιχτών δεδομένων διαδραματίζει το νομικό καθεστώς που θα θεσπίσουν οι κυβερνήσεις γύρω από τα ανοιχτά δεδομένα, ώστε να αισθάνονται οι πολίτες ασφαλείς αλλά και ηθικά δίκαιοι. Ήδη έχουν αναφερθεί αρκετές παρεμβάσεις σε χώρες όπως για παράδειγμα η Ινδία. Οι παρεμβάσεις αυτές αφορούν κυρίων τα δικαιώματα των πολιτών έναντι του κράτους.

Το πρώτο βήμα είναι η κάθε κυβέρνηση να προσπαθήσει να οικοδομήσει ένα πλαίσιο εμπιστοσύνης με τους πολίτες της. Μέσα σε αυτό το πλαίσια θα πρέπει οι πολίτες να αισθάνονται ταυτόχρονα ελεύθεροι και ασφαλείς. Για παράδειγμα η αυστραλιανή κυβέρνηση έχει θεσπίσει ειδικό νόμο περί απορρήτου όπου προσωπικές ταυτοποιητικές πληροφορίες των πολιτών όπως ονόματα και διευθύνσεις αποκρύπτονται από τα κυβερνητικά δεδομένα προτού αυτά δημοσιοποιηθούν (Hardy, Maurushat, 2017). Σε καμία περίπτωση δεν ενδείκνυται μία καταναγκαστική συλλογή δεδομένων χωρίς πρωτίστως την συγκατάθεσή των πολιτών. Μία τέτοια περίπτωση θα έφερνε τα αντίθετα αποτελέσματα αφού οι πολίτες θα έπαυαν να νιώθουν ελεύθεροι κάτω από το νέο καθεστώς. Και ενώ υπάρχουν δεδομένα που εξαρτώνται άμεσα από την κυβέρνηση όπως πχ τα μητρώα των πολιτών υπάρχουν κάποια άλλα δεδομένα που οι πολίτες θα έπρεπε να διαμοιραστούν με την κυβέρνηση. Σε περιπτώσεις που οι πολίτες δεν νιώθουν εμπιστοσύνη στην κυβέρνηση, έγκειται πάντα η αμφιβολία της αξιοπιστίας των δεδομένων. Ένα απτό παράδειγμα ήταν η απογραφή του 2021 που πραγματοποιήθηκε στη χώρα μας, όπου οι πολίτες κλήθηκαν να απαντήσουν οι ίδιοι και ειλικρινά σε διάφορα ερωτήματα της κυβέρνησης.

Παρόμοια αντιμετώπιση μπορεί να συμβεί και σε περιπτώσεις που οι πολίτες δεν νιώθουν ασφάλεια των προσωπικών τους δεδομένων. Υπάρχουν αρκετά παραδείγματα κυρίως με κακόβουλες επιθέσεις εναντίων εταιριών με σκοπό την διαρροή δεδομένων. Μετά από αντίστοιχα συμβάντα που έχουν συμβεί έχει παρατηρηθεί έντονα το φαινόμενο, οι πολίτες να διστάζουν να δώσουν τα αληθινά προσωπικά στοιχεία.

Όταν λοιπόν δεν υπάρχει το κατάλληλο πλαίσιο να νιώθουν οι πολίτες ασφάλεια και εμπιστοσύνη στην κυβέρνηση τους, τότε και παρατηρείται χειραγώγηση των δεδομένων, μέσω για παράδειγμα ψευδώς δεδομένων και πληροφοριών. Αυτό μπορεί να τους δώσει την

δύναμη να ελέγξουν και να επηρεάσουν τον τρόπο διακυβέρνησης των κρατών τους, και για αυτό οι κυβερνήσεις θα πρέπει να είναι πολύ προσεκτικές στο χειρισμό του φαινομένου. Μπορεί δηλαδή η κυβέρνηση να σχεδιάσει τρόπους και επεξεργασίας των δεδομένων και να ληφθούν αποφάσεις με γνώμονα τα αναληθή δεδομένα των πολιτών.

2.3.4 Πηγές πληροφόρησης

Άλλο ένα στοιχείο που χρειάζεται να αντιμετωπιστεί πριν την θέσπιση των ανοιχτών δεδομένων είναι η ύπαρξη των κρυφών και απόρρητων εγγράφων. Μεγάλο μέρος αυτών των εγγράφων παραμένουν απόρρητα και άγνωστα προς τους πολίτες καθώς το ίδιο το κράτος δεν είναι σε θέση να αντιμετωπίσει τις συνέπειες και τα προβλήματα που μπορεί να επιφέρει η δημοσίευση τέτοιων πληροφοριών. Εδώ ξανά έρχεται η σημασία του να προαποφασιστεί σε ποιο βαθμό θα υπάρξουν τα ανοιχτά δεδομένα.

Ένα ακόμη σημαντικό θέμα είναι πως το σύνολο των δεδομένων συχνά προέρχεται από πολλαπλές υπηρεσίες. Η συλλογή αλλά και η ενοποίηση όλων αυτών των στοιχείων έχει αποδειχθεί ότι είναι μία χρονοβόρα και δύσκολη διαδικασία. Επίσης ακόμη και αν ανακτηθούν υπάρχει η πιθανότητα να μην παρέχουν έγκυρες πληροφορίες ή και ακόμη να αλληλοαναιρούνται στοιχεία προερχόμενα από διαφορετικές πηγές. Αυτό συμβαίνει γιατί εντοπίζονται πληροφορίες και δεδομένα που έχουν καταγραφεί ανεπίσημα ή έγιναν αποδεκτά χωρίς να πιστοποιηθεί η εγκυρότητα τους. Πολλές φορές μάλιστα παρατηρείται τα ίδια τα δεδομένα να μην καταγράφονται με δομημένη μορφή, γεγονός που κατά την ανάκτηση τους δυσκολεύει την διαδικασία της επεξεργασίας τους. Έτσι, ως συνέπεια υπάρχει πάντα ο κίνδυνος να ληφθούν αποφάσεις όχι βασιζόμενες στο πραγματικό σύνολο των δεδομένων που υπάρχουν αλλά σε ένα υπό σύνολο των δεδομένων τα οποία έχουν εντοπιστεί με δομημένη μορφή. Συχνά, ωστόσο έχει παρατηρηθεί τα δύο αυτά σύνολα μπορεί να διαφέρουν σημαντικά μεταξύ τους. Μάλιστα υπάρχει πιθανότητα και να οδηγούν σε διαφορετικά αποτελέσματα και συμπεράσματα.

Έτσι όλες οι πλέον ανεπτυγμένες χώρες, αλλά και αρκετές αναπτυσσόμενες έχουν στραφεί προς αυτήν την κατεύθυνση, προσπαθώντας να περιορίσουν τα προβλήματα των πολλαπλών πηγών πληροφόρησης στο βαθμό που αυτό είναι δυνατό.

2.3.5 Συμπέρασμα

Τα παραπάνω ζητήματα θα πρέπει να αντιμετωπιστούν με μεγάλη προσοχή και μεθοδικότητα από τις κυβερνήσεις. Διαφορετικά θα υπάρχει η ανησυχία για το τι μπορεί να επιφέρει τελικά μία στρατηγική ανοιχτών δεδομένων, όπου εκτός από τα πλεονεκτήματα υπάρχουν και αρνητικά οφέλη. (Yu and Robinson, 2012). Θα πρέπει λοιπόν οι ίδιες οι κυβερνήσεις πριν χρησιμοποιούν τα δεδομένα ως εργαλείο πολιτικής να έχουν κατανοήσει τον τρόπος λειτουργίας τους αλλά και όλων των κινδύνων που συνυπάρχουν. Η κάθε κυβέρνηση θα που θέλει να λειτουργήσει με ΑΚΔ θα πρέπει να λάβει αποφάσεις ως προς τον βαθμό που θα έχουν ανοιχτά δεδομένα, με ποιες διαδικασίες θα γίνει η συλλογή και η επιλογή των δεδομένων καθώς και τις πλατφόρμες που θα λάβει χώρα η αποθήκευσή τους.

Σημαντική συμβολή στην επίλυση των ανωτέρω ζητημάτων φαίνεται πως είναι η ίδρυση μιας διαχειριστικής ομάδας εμπειρογνωμόνων. Αυτή η ομάδα θα είναι υπεύθυνη για την διαχείριση, τη διαλογή των δεδομένων και τη επεξεργασία των κρατικών δεδομένων. Στις αρμοδιότητες της συγκεκριμένης κυβερνητικής ομάδας θα είναι και ο έλεγχος αξιοπιστίας των δεδομένων, αλλά και η αποταυτοποίηση τους από τις προσωπικές απόρρητες πληροφορίες των πολιτών. Είναι βασικό τα δεδομένα που συλλέγονται να μην είναι προσωποποιημένα, καθώς έτσι οι πολίτες θα αισθάνονται ότι η συλλογή των πληροφορίων γίνεται για να ελέγχεται η προσωπική τους ζωή. Σε ένα δημοκρατικό καθεστώς, όπου διέπετέ από ευχέρεια λόγου και κινήσεων, κάτι τέτοιο δεν θα γίνει αποδεκτό. Έτσι η συγκεκριμένη ομάδα που θα συσταθεί για την συλλογή των δεδομένων θα πρέπει να απαρτίζεται από ανθρώπους με γνώσεις τεχνολογίας και ψηφιακών λύσεων (Hardy, Maurushat, 2017). Πέραν του κοινού μορφωτικού επιπέδου θα πρέπει οι άνθρωποι της ομάδας διαχείρισης των δεδομένων να προέρχονται από ποικίλες κοινωνικές ομάδας, φύλλο, φυλή, ακόμη και σεξουαλικό προσανατολισμό ώστε να μπορέσει να δοθεί αντικειμενικότητα στα δεδομένα (Barryhill, Heang, Clogher, McBride, 2019). Τέλος εξίσου σημαντικό ρόλο διαδραματίζει η

ίδια η κοινωνία και η αποδοχή και πλήρης κατανόηση των ανοιχτών κυβερνητικών δεδομένων. Σε αυτό το πλαίσιο ανήκει και οι συζητήσεις που έχουν αρχίσει να γίνονται περί αντίστοιχης εκπαίδευση στα σχολεία, με προτάσεις όπως το να εισαχθεί στον σχολικό πρόγραμμα σπουδών η επιστήμη των δεδομένων.

Σύμφωνα με τα παραπάνω προκύπτει ως λογικό συμπέρασμα ότι τα εργαλεία και οι αλγόριθμοι με τα οποία γίνεται η ανάλυση των δεδομένων δεν είναι από μόνα τους αμερόληπτα και αληθή επειδή απλώς βασίζονται σε δεδομένα. Αντιθέτως το είδος και η ποιότητα των δεδομένων είναι αυτό που θα καθορίσει αν το κάθε εργαλείο αποτυπώνει την πραγματικότητα αμερόληπτα. Το ίδιο το εργαλείο όμως (ο αλγόριθμός για παράδειγμα) δεν είναι εύκολα κατανοητός από το σύνολο του πληθυσμού. Αντίθετα τα δεδομένα είναι τα θεμελιώδη στοιχεία για την μηχανική μάθηση. Για αυτό οι κυβερνήσεις που θα θελήσουν να ακολουθήσουν αντίστοιχη στρατηγική θα πρέπει να έχουν πρόσβαση σε αληθή και χρήσιμα δεδομένα, τα οποία θα προστατεύουν των προσωπική ζωή των πολιτών και θα συνάδουν με τους ηθικούς κανόνες..

2.4 Ανοιχτά δεδομένα στη Σκωτία

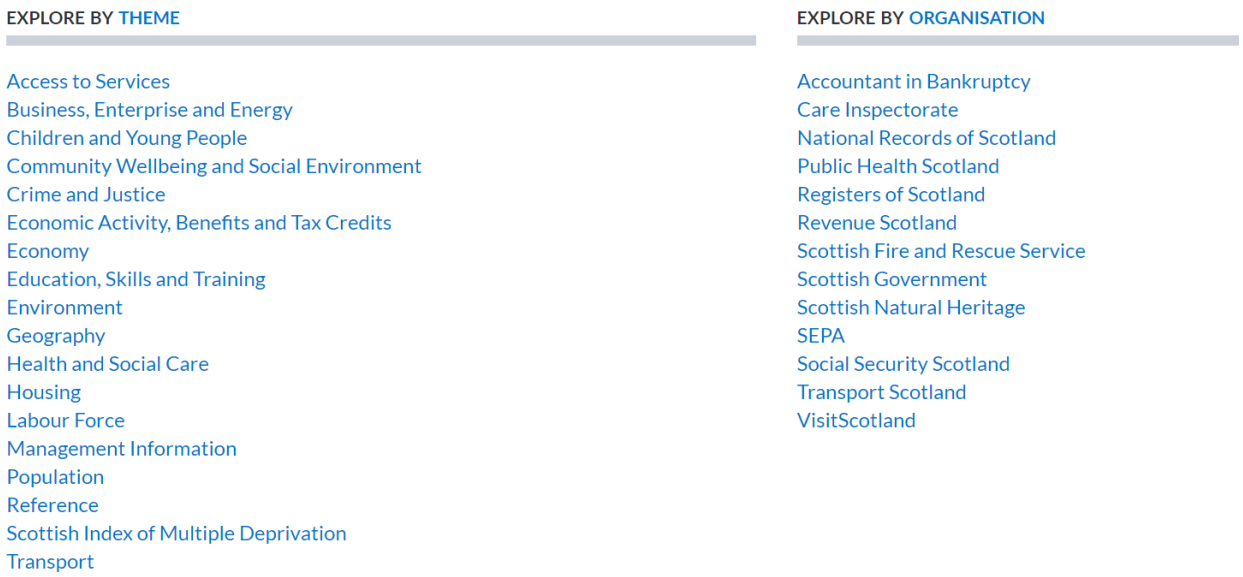
2.4.1 Linked Open Data

Η κυβέρνηση της Σκωτίας ασκεί μία συγκεκριμένη στρατηγική ανοιχτών κυβερνητικών δεδομένων όπου δημοσιεύει μεγάλη συλλογή επίσημων στατιστικών δεδομένων ως Linked Data. Με τον όρο Linked data ορίζονται οι σχέσεις και οι συνδέσεις μεταξύ των δεδομένων από διάφορες πηγές δεδομένων όπως για παράδειγμα από βάσεις δεδομένων και από το Web (Cai, Vasilakos, 2017). Τα ανοιχτά συνδεδεμένα δεδομένα (Linked Open Data) παρέχουν πρόσβαση σε δομημένα δεδομένα τα οποία διέπονται από τους κανόνες του Σημασιολογικού ιστού (Semantic Web). Σύμφωνα με τον Tim Berners-Lee με τον όρο Semantic Web δεν εννοεί απλά την αποθήκευση των δεδομένων, αλλά και τη δημιουργία συνδέσμων μεταξύ τους ώστε να μπορεί να τα εξερευνήσει ένας άνθρωπος ή μία μηχανή (Meymandpour, Joseph G.Davis, 2015).

Όπως όλες οι χώρες έτσι και η κυβέρνηση της Σκωτίας συγκεντρώνει μεγάλο όγκο δεδομένων για διάφορους τομείς, και τα στοιχεία αυτά δημοσιεύονται στο statistics.gov.scot. Η επίσημη σελίδα των στατιστικών δεδομένων της Σκωτίας δημοσιεύτηκε στο ευρύ κοινό το Φεβρουάριο του 2016, και τα στοιχεία που παρέχει προέρχονται από διάφορους οργανισμούς, με κύριο την ίδια την κυβέρνηση της Σκωτίας, το εθνικό αρχείο (National Records of Scotland), και το NHS Information Services Division and Transport Scotland. Τα φάσμα των στατιστικών δεδομένων ποικίλει, και αφορά στοιχεία όπως οικονομία στοιχεία, φορολογία, εκπαίδευση, εγκληματικότητα, ιατρική φροντίδα και άλλα, και σχεδόν για κάθε σύνολο υπάρχει η διάσταση της γεωγραφικής περιοχής και του χρόνου.

2.4.2 Περιήγηση στη σελίδα

Η κυβέρνηση τη Σκωτίας, όπως έχει ήδη αναφερθεί, διαθέτει ανοιχτά κυβερνητικά δεδομένα. Στην επίσημη σελίδα όπου γίνεται η ανάκτηση των στοιχείων έχουν ήδη κοινοποιηθεί πάνω από 300 datasets. Στην αρχική σελίδα εμφανίζεται ένα search bar, όπου μπορεί κανείς να αναζητήσει κάποια συγκεκριμένο dataset ή μπορεί να επιλέξει να δει όλα τα dataset ανά θέμα ή ακόμη και ανά οργανισμό (πηγή συλλογής δεδομένων).



Εικόνα 1: Απεικόνιση της ιστοσελίδας ΑΚΔ της Σκωτίας

Μετά από κάθε επιλογή του στοιχείου δεδομένων μπορεί να εισαχθεί η γεωγραφική διάσταση αλλά και η διάσταση του χρόνου. Η γεωγραφική διάσταση σχετίζεται με τις διοικητικές περιοχές. Η Σκωτία είναι αρχικά χωρισμένη σε 32 περιοχές που ονομάζονται Council areas, αλλά και σε μικρότερες περιοχές τις Data Zones. Αρχικά υπήρχαν οι 2001 Data Zones, οι οποίες βασίζονται στην απογραφή του 2001, αλλά μετά τη απογραφή του 2011 διαμορφώθηκαν οι 2011 Data Zones. Πρόκειται για μικρές γεωγραφικές, οι οποίες έχουν σχετικά παρόμοιο πληθυσμό, αλλά μπορεί να διαφέρει το μέγεθος της έκτασης της περιοχή. Στο σύνολο τους υπάρχουν 6.976 2011 Data Zone.

Αναλόγως με τη γεωγραφική περιοχή μπορούν να επιλεγούν και η διάσταση του χρόνου. Για παράδειγμα καθώς η γεωγραφική περιοχή 2011 Data Zone δεν υπήρχε πριν το 2011, και για αυτό δεν μπορούν να επιλεγούν στατιστικά στοιχεία για μία τέτοια περιοχή σε προγενέστερο έτος από του 2011. Αντιστοίχως τα δεδομένα που αναφέρονται σε έτος πριν το 2011 η μικρότερη δυνατή γεωγραφική περιοχή που μπορεί να επιλεγεί ως διάσταση είναι η 2011 Data Zones.
Στη συνέχεια από επιλεγούν τα δεδομένα που επιθυμεί ο χρήστης, μπορεί είτε να γίνει η απευθείας χρήσης τους με API, είτε και να κατέβουν τοπικά πχ σε excel ή txt μορφή.

3. Θεωρητικό Υπόβαθρο

3.1 Επιστήμη των δεδομένων

3.1.1 Μηχανική Μάθηση

Η εμφάνιση αλλά και ανάπτυξη των ψηφιακών δεδομένων αλλά κυρίως των μεγάλων δεδομένων (big data) είναι οι κύριοι λόγοι άνθισης της επιστήμης των δεδομένων. Η επιστήμη των δεδομένων χρησιμοποιεί τις θεωρίες και τις μεθόδους από ένα μεγάλο φάσμα επιστημών όπως είναι τα μαθηματικά, η στατιστική και η επιστήμη των υπολογιστών. Με τη χρήση της επιστήμης των δεδομένων μπορούν να ερμηνευτούν τα δεδομένα και τα παραχθεί πληροφορία πολύτιμη για την λήψη των αποφάσεων. Πρόκειται στην ουσία για την θεωρία

αλλά και τη μεθοδολογία σύμφωνα με τις οποίες θα μετατραπούν τα δεδομένα σε πληροφορίες, και αντίστοιχα οι πληροφορίες σε γνώση (Xu, Tang, Xu, Cheng, 2021)

Η επιστήμη των δεδομένων ενώ προϋπήρχε από το 1960, ξεκίνησε να αναπτύσσεται μετά το 1990, και μάλιστα γνώρισε ιδιαίτερη ανάπτυξη με την διέλευση των μεγάλων δεδομένων (big data). Η επιστήμη των δεδομένων είναι ένα γενικό επίπεδο που περιλαμβάνει την τεχνητή νοημοσύνη αφού ο πυρήνας της είναι οι αλγόριθμοι (Xu, Tang, Xu, Cheng, 2021). Η τεχνητή νοημοσύνη είναι ο τομέας που βασίζεται στον τρόπο λειτουργίας των ανθρώπινων διαδικασιών και συμπεριφορών αλλά και τον τρόπο σκέψης για την λήψη των αποφάσεων. Η τεχνητή νοημοσύνη χρησιμοποιεί διαφορετικές μαθηματικές προσεγγίσεις από αυτήν της επιχειρησιακής έρευνας, αφού βασίζεται περισσότερο σε εξόρυξη δεδομένων (data mining), αναγνώρισης προτύπων (pattern recognition), αλλά και μηχανικής μάθησης (machine learning) (Gruson, Hellepute, Rousseau, Gruson, 2019). Η μηχανική μάθηση είναι στην ουσία υπό πεδίο της τεχνητής νοημοσύνης η οποία βασίζεται επίσης στην στατιστική αλλά με σκοπό την εύρεση μοτίβων, τάσεων και συσχετίσεων.

Ο τομέας της μηχανικής μάθησης (Machine Learning) επεκτείνεται πλέον με ταχύτερους ρυθμό αύξησης. Οι λόγοι για την ακμάζουσα πορεία του είναι κυρίως η διέλευση των μεγάλων δεδομένων, το χαμηλό υπολογιστικό κόστος αλλά και η ανάπτυξη νέων θεωριών γύρω από τους αλγόριθμους (Pugliese, Regondi, Marini, 2021). Με τον όρο Μηχανική Μάθηση ορίζονται οι υπολογιστικές μεθόδους οι οποίες χρησιμοποιούν την εμπειρία με σκοπό να βελτιωθεί η απόδοση ή ακόμη και να κάνουν ακριβείς προβλέψεις (Mohri, Rostamizadeh, Talwalkar, 2012). Με την έννοια της εμπειρίας εννοείται οι προηγούμενες πληροφορίες που υπάρχουν διαθέσιμες. Με την μηχανική μάθηση λοιπόν σχεδιάζονται αποτελεσματικοί αλγόριθμοι πρόβλεψης οι οποίοι βασίζονται στις προηγούμενες πληροφορίες που λαμβάνουν. Κυρίως δηλαδή αναφέρεται στην δημιουργία αλλά και την έρευνα για τον κατάλληλο τύπου αλγορίθμου ο οποίος θα μπορέσει να μάθει από τα δεδομένα και να είναι σε θέση να παράγει ακριβείς προβλέψεις (Xu, Tang, Xu, Cheng, 2021). Ωστόσο παρά την ακμάζουσα πορεία και το συνεχές ενδιαφέρον που φαίνεται να αποκτά η μηχανική μάθηση θεωρείται ότι βρίσκεται ακόμη στην αρχή και ότι έχει ακόμη αμέτρητες δυνατότητες και ερευνητικά πεδία ενδιαφέροντος (Pugliese, Regondi, Marini, 2021).

Ένα βασικό μειονέκτημα της παραδοσιακής μηχανικής μάθησης είναι ότι θα πρέπει πριν την διαδικασία της εκμάθησης και την εφαρμογή του αλγορίθμου να γίνεται η επιλογή των χαρακτηριστικών (feature extraction). Αυτά τα χαρακτηριστικά δεν μπορούν να εξαχθούν αυτόματα αλλά αντιθέτως θα πρέπει να καθορίζονται χειροκίνητα. Το συγκεκριμένο μειονέκτημα έρχεται να το επιλύσει η βαθιά μηχανική μάθηση (deep machine learning), η οποία ουσιαστικά είναι και μία μέθοδος εκμάθησης χαρακτηριστικών. Με την χρήση της βαθιάς μηχανικής μάθησης τα δεδομένα αποκτούν πιο σύνθετες και αφηρημένες έννοιες ώστε να μπορέσει να επιτευχθεί η αυτόματη εξαγωγή χαρακτηριστικών αλλά και οι διαδικασίες της εκμάθησης (Xu, Tang, Xu, Cheng, 2021).

3.1.2 Διαδικασίες Μάθησης

Οι αλγόριθμοι της μηχανικής μάθησης μπορούν να ταξινομηθούν σε τέσσερις βασικές κατηγορίας ανάλογα με τα είδος της μάθησης που παρέχουν. Κυρίως γνώμονας για την επιλογή της διαδικασία μάθησης είναι τα δεδομένα που θα χρησιμοποιηθούν από το μοντέλο. Το είδος των δεδομένων αναφέρεται τόσο στα δεδομένα που θα χρησιμοποιηθούν στην διαδικασία της εκπαίδευσης του μοντέλου (training data), αλλά και σε αυτά που θα χρησιμοποιηθούν για την αξιολόγηση του αλγορίθμου εκμάθησης (test data) (Mohri, Rostamizadeh, Talwalkar, 2012). Έτσι λοιπόν σύμφωνα με τα data που χρησιμοποιούνται έχουν δημιουργηθεί οι παρακάτω τέσσερις βασικές κατηγορίες: η εποπτευόμενη μάθηση (supervised learning), η ημι - εποπτευόμενη μάθηση (semi - supervised learning), η μάθηση χωρίς επίβλεψη (unsupervised learning), και η ενισχυτική μάθηση (reinforcement learning).

Εποπτευόμενη μάθηση (supervised learning)

Η εποπτευόμενη μάθηση αποτελεί τον πιο συνηθισμένο τρόπο εκμάθησης, και ονομάζεται έτσι καθώς στο σύνολο των δεδομένων που παρέχονται για εκπαίδευση έχει επισημανθεί η αναζητούμενη μεταβλητή με ετικέτα (label). Κατά τη διαδικασία της εκπαίδευση γίνεται σύγκριση της προβλεπόμενης μεταβλητής με την υπολογιζόμενη μεταβλητή, και υπολογίζεται αντίστοιχα το σφάλμα. Όσο διαρκεί η εκπαίδευση γίνεται η προσαρμογή του σφάλματος με στόχο την επίτευξη του ελάχιστου δυνατού σφάλματος (Mohri,

Rostamizadeh, Talwalkar, 2012). Χαρακτηριστικά παραδείγματα εποπτευόμενης μάθησης είναι ο εντοπισμός των ανεπιθύμητων μηνυμάτων (spam), αλλά και η ανίχνευση προσώπου (face recognition) (Pugliese, Regondi, Marini, 2021).

Μάθηση χωρίς επίβλεψη (unsupervised learning)

Τα μοντέλα που εκπαιδεύονται χωρίς επίβλεψη δεν έχουν καθόλου ετικέτα στα δεδομένα της εκπαίδευσης τους (train data), καθώς είναι δύσκολο να δημιουργηθεί. Αυτό βέβαια δημιουργεί αντίστοιχη δυσκολία στον τρόπο αξιολόγησης του μοντέλου (Mohri, Rostamizadeh, Talwalkar, 2012). Η πιο συνηθισμένη χρήση αυτών των αλγορίθμων είναι να ταξινομεί τα δείγματα σε ομάδες, όπως για παράδειγμα στην αναγνώριση φίλων ενός χρήστη στα μέσα κοινωνικής δικτών (social media) (Pugliese, Regondi, Marini, 2021).

Ημι - εποπτευόμενη μάθηση (semi - supervised learning)

Η ημι εποπτευόμενη μάθηση είναι ένας συνδυασμός της εποπτευόμενης και της μάθησης χωρίς επίβλεψης, αφού στο μοντέλο παρέχονται για εκπαίδευση και δεδομένα με ετικέτα αλλά και δεδομένα χωρίς ετικέτα. Στόχος αυτών των μοντέλων είναι να βελτιώσουν την πρόβλεψη που μπορούσαν να προσφέρουν οι αλγόριθμοι που θα είχαν μόνο δεδομένα με ετικέτα (Pugliese, Regondi, Marini, 2021). Χρησιμοποιείται ευρέως σε εφαρμογές που είναι δύσκολο και ακριβό να δημιουργηθεί ετικέτα για όλα τα δεδομένα, με χαρακτηριστικό παράδειγμα αυτού του τρόπο μάθησης είναι η μηχανές μετάφρασης και η ταξινόμηση κειμένου (Mohri, Rostamizadeh, Talwalkar, 2012).

Ενισχυτική μάθηση (reinforcement learning)

Η ενισχυτική μάθηση είναι ένα είδος μάθησης όπου εναλλάσσονται οι διαδικασίες της εκπαίδευσης (training phase) και της αξιολόγησης (testing phase) και αξιολογείται η βέλτιστη συμπεριφορά μέσα σε ένα περιβάλλον. Το μοντέλο αλληλοεπιδρά με το περιβάλλον, και μαθαίνει για την κάθε του ενέργεια μέσω ανταμοιβής ή τιμωρίας. Αντίστοιχα παραδείγματα ενισχυτικής μάθησης εφαρμόζονται στα συστήματα της ρομποτικής όπως η αυτόνομη οδήγηση (Pugliese, Regondi, Marini, 2021).

3.1.3 Είδη Προβλημάτων Μηχανικής Μάθησης

Αντίστοιχα ανάλογα με την πρόβλεψη του θα γίνει από το μοντέλο, τα προβλήματα που προσπαθεί να αντιμετωπίσει ένας αλγόριθμός μπορούν να κατηγοριοποιηθούν σε δύο βασικές κατηγορίες. Βασικό κριτήριο αυτήν τη φορά δεν είναι τα δεδομένα που δέχεται κατά την εκπαίδευση, αλλά το είδος της μεταβλητής που προσπαθεί να προβλέψει.

Κατηγοριοποίηση (Classification)

Τα προβλήματα κατηγοριοποίησης (ή ταξινόμησης) αφορούν την εκμάθηση μιας συνάρτησης - ταξινομητής. Η εκπαίδευση γίνεται με ένα συγκεκριμένο σύνολο δεδομένων (training set), όπου κάθε μεταβλητής *x* μπορεί να έχει *n* χαρακτηριστικά, και ανήκει σε μία κλάση (ομάδα) *y*. Σκοπός ενός προβλήματος κατηγοριοποίησης είναι να βρεθεί η κατάλληλη mapping συνάρτηση όπου να κατηγοριοποιεί την ανεξάρτητη μεταβλητή *x* σε μία κλάση *y*, με το λιγότερο δυνατό σφάλμα. Ένα κλασικό παράδειγμα classification είναι ο διαχωρισμός των email σε spam. Το μοντέλο εκπαιδεύεται με διάφορους τύπους email, και στην συνέχεια μπορεί να προβλέψει αν ένα νέο εισερχόμενο email θα θεωρηθεί ως spam και θα ταξινομηθεί στον spam φάκελο ή όχι. Ένα πολύ συνηθισμένο ζήτημα που συναντάται στα προβλήματα κατηγοριοποίησης είναι η ανισορροπία των δεδομένων. Αυτό σημαίνει ότι η πλειοψηφία των δεδομένων του training dataset ανήκουν σε μία κλάση, με αποτέλεσμα η πρόβλεψη της συνάρτησης να δίνει ως επι των πλείστων την κλάση στην οποία ανήκουν η πλειονότητα των χαρακτηριστικών (Sanz, Sesma-Sara, Bustince, 2021)

Παλινδρόμηση (Regression)

Τα regression προβλήματα αφορούν την εύρεση των συσχετίσεων μεταξύ των ανεξάρτητων και των εξαρτημένων μεταβλητών. Σκοπός είναι να βρεθεί η βέλτιστη συνάρτηση, όπου θα χρησιμοποιεί τα χαρακτηριστικά των μεταβλητών *x* και προσπαθεί να προβλέψει την τιμή μιας συνεχούς μεταβλητής *y* με το ελάχιστο σφάλμα (Dawoud, Lukman, Haadi, 2021). Χαρακτηριστικό τέτοιο παράδειγμα είναι η πρόβλεψη των τιμών των μετοχών. Έχοντας ως ανεξάρτητες μεταβλητές τις προηγούμενες κινήσεις της μετοχής, η συνάρτηση προσπαθεί να προβλέψει την επόμενη τιμή που θα λάβει η μετοχή.

3.2 Αλγόριθμός XGB

3.2.1 EXtreme Gradient Boosting

Στην μηχανική μάθηση έχουν δημιουργηθεί ποικίλοι αλγόριθμοι αλλά και παραλλαγές αυτών. Ένας αλγόριθμός ο οποίος χρησιμοποιείται μάλιστα αρκετά συχνά και στις βιβλιογραφικές έρευνες αλλά και στην πράξη είναι ο gradient tree boosting. Η χρήση του προορίζεται κυρίως για δομημένων δεδομένα σε μορφή πινάκων (tabular data), και έχει γίνει ιδιαίτερα δημοφιλής μέσα από τους διαγωνισμούς του kaggle λόγω της ταχύτητας και της ακρίβειας που προσφέρει. Ο συγκεκριμένος αλγόριθμος πρόκειται ουσιαστικά για μία εφαρμογή των δέντρων αποφάσεων (decision tree).

Το μοντέλο των δέντρων αποφάσεων (decision tree) είναι ένα διάγραμμα ροής που μοιάζει με τη δομή του δέντρο. Στο δέντρο απόφασης κάθε εσωτερικός κόμβος (node) συμβολίζεται με ορθογώνιο, ενώ τα φύλλα (leaf nodes) αναπαρίστανται με οβάλ σχήμα. Κάθε εσωτερικός κόμβος αντιστοιχεί και σε μία παραδοχή, και διαχωρίζεται σε δύο ή περισσότερους κόμβους αναλόγως με την τιμή που θα πάρει. Αντίστοιχα κάθε φύλλο είναι ένα αποτέλεσμα, μία ετικέτα κλάσης δηλαδή (label), και ο κάθε κλάδος αντιπροσωπεύει ένα πιθανό σενάριο απόφασης (Yadav, Pal, 2012). Έτσι λοιπόν ξεκινώντας από τον πρώτο κόμβο (root node), και αναλόγως την τιμή που θα λάβει, θα καταλήξει στη συνέχεια στο επόμενο κόμβο. Αυτό θα συμβαίνει μέχρι να φτάσει στο τελικό αποτέλεσμα.

Το boosting είναι η τεχνική της συνεχόμενης προσθήκης νέων μοντέλων με σκοπό την διόρθωση των σφαλμάτων. Τα μοντέλα συνεχίζουν τα προστίθενται διαδοχικά μέχρι το σημείο που η επιπλέον προσθήκη δεν επιφέρει περαιτέρω διόρθωση του σφάλματος. Με την τεχνική του Gradient boosting τα νέα μοντέλα που προστίθενται προβλέπουν τα κατάλοιπα (σφάλματα) των προηγούμενων μοντέλων και συνδυάζονται ώστε να παραχθεί η τελική πρόβλεψη. Βασίζονται στον αλγόριθμο gradient descent σκοπός του οποίου είναι να ελαχιστοποιηθεί η απώλεια (loss) από την προσθήκη των νέων μοντέλων.

Το boosting ξεκίνησε από τον Schapire το 1990, σύμφωνα με τον οποίο ακόμη και ένα αδύναμο μοντέλο εκμάθησης μπορεί πάντα να βελτιώσει την απόδοση του προσθέτοντας δύο επιπλέον μοντέλα. Το 1995 ο Freund πρότεινε μία βελτιωμένη εκδοχή του αλγορίθμου του Schapire, τον "Boost of majority". Σύμφωνα με τον Feund η απόδοση ενός αλγορίθμου μπορεί να βελτιωθεί συνδυάζοντας ταυτόχρονα πολλούς αδύναμους εκπαιδευτές. Ωστόσο και οι δυο απόψεις βασιζόταν στη θεωρία ότι classifier θα πρέπει να έχει ένα σταθερό ποσοστό σφάλματος, η οποία απορρίφθηκε το 1996 από τους Schapire και Freud και έτσι δημιουργήθηκε ο Adaptive boosting αλγόριθμός (AdaBoost) (Friedman, Hastie, Tibshirani, 2000).

Ο gradient boosting decision tree είναι ένα αλγόριθμος επαναλαμβανόμενων δέντρων αποφάσεων (decision tree). Σε κάθετου επανάληψη τα μοντέλα εκπαιδεύονται χρησιμοποιώντας τα κατάλοιπα των προηγούμενων δέντρων αποφάσεων. Ως τελικό αποτέλεσμα προκύπτει το συσσωρευμένο άθροισμα όλων των δέντρων. Ο αλγόριθμός Extreme Gradient Boosting, βασίζεται σε ένα συγκεκριμένο είδος δέντρων αποφάσεων (decision tree), το CART. Η ειδοποιός διαφορά ενός δέντρου CART είναι ότι κάθε φύλλο (leaf) δεν αντιστοιχεί απαραιτήτως απλώς σε μία τιμή (αποτέλεσμα), αλλά έχει και το βάρος του (score). Μπορεί να χρησιμοποιηθεί και σε περιπτώσεις classification, όπου το τελικό φύλλο θα οδηγεί σε μία κλάση (class), αλλά και σε περιπτώσεις regression, όπου θα αναζητείται μία συνεχής μεταβλητή. Οι παράμετροι που χρησιμοποιούνται στον αλγόριθμό του Gradient boost είναι ο αριθμός των επαναλήψεων (interactive), ο ρυθμός εκμάθησης (learning rate), το βάθος των δέντρων (maximum depth of trees), η αντικειμενική συνάρτηση (objective function) (Gu, Chang, Xiong, Chen, 2020).

Αντικειμενική συνάρτηση (objective function)

Η αντικειμενική συνάρτηση περιλαμβάνει την συνάρτηση απώλειας (loss function), η οποία χρησιμοποιείται για να αναπαραστήσει το σφάλμα απώλειας. Σκοπός είναι να ελαχιστοποιηθεί το σφάλμα της αντικειμενικής συνάρτησης, το οποίο προκύπτει από την σύγκριση της πραγματικής τιμής και της προβλεπόμενης τιμής. Κατά την εκπαίδευση ενός XGB μοντέλου, κάθε επιπλέον "αδύναμος" εκπαιδευτής που προστίθεται προσπαθεί να μειώσει το σφάλμα απώλειας των προηγουμένων επαναλήψεων. Το είδος της αντικειμενικής

συνάρτησης απώλειας εξαρτάται από την κατηγορία του προβλήματος, αν πρόκειται δηλαδή για regression ή classification. Για παράδειγμα σε ένα πρόβλημα regression συνήθως επιλέγεται η μέθοδος ελαχίστων τετραγώνων (Rostamian, Heidaryan, Ostadhassan, 2021)

Αριθμός των επαναλήψεων k (interactive)

Ο αριθμός k εκφράζει τον αριθμό των δέντρων που θα χρησιμοποιηθούν στο μοντέλο, όπου το άθροισμα των αποτελεσμάτων του κάθε δέντρου θα χρησιμοποιηθεί στο τελικό αποτέλεσμα. Όταν ο αριθμός των επαναλήψεων φτάσει στην μέγιστη τιμή που του έχει δοθεί τότε το αποτέλεσμα θα είναι οι βέλτιστες τιμές που έλαβε κατά τη διάρκεια της εκπαίδευσης.

Maximum depth of trees

Το μέγεθος των δέντρων που θα επιλεχθεί ως παράμετρος επηρεάζει την απόδοση του μοντέλου. Με το όρο depth of tree εννοείται η απόσταση που έχει ένας κόμβος (node) μέχρι να φτάσει στον κύριο κόμβο (root node). Έτσι όσο μεγαλύτερο είναι το βάθος του δέντρου σημαίνει ότι τόσο περισσότεροι κόμβοι υπάρχουν μέχρι να τον root node. Ένα δέντρο αποφάσεων με υψηλό βάθος θα έχει καλύτερη απόδοση, ωστόσο κινδυνεύει να αυξηθεί η πολυπλοκότητά του. Εάν έχει υπερβολικά μεγάλο αριθμό, τότε το μοντέλο μπορεί να έχει σφάλματα υπερ προσαρμογής (overfitting), δηλαδή να έχει ιδιαίτερη καλή προσαρμογή στα training δεδομένα αλλά μην μπορέσει να κάνει προβλέψεις στα test δεδομένα.

Learning rate

Ο ρυθμός εκμάθησης (learning rate) καθορίζει το μέγεθος του βήματος που θα κάνει κάθε φορά το μοντέλο, σε κάθε επανάληψη του. Εάν λάβει μία πολύ μικρή τιμή το μοντέλο θα μαθαίνει αργά αλλά σταθερά, και θα χρειαστεί περισσότερες επαναλήψεις. Συνήθως με μικρές τιμές στο ρυθμό εκμάθησης επιτυγχάνονται καλύτερα αποτελέσματα, ωστόσο το μοντέλο κινδυνεύει και πάλι από overfitting.

3.3 Βαθιά Μηχανική Μάθηση

3.3.1 Εισαγωγή

Μία πολύ διαδεδομένη κατηγορία μηχανικής μάθησης είναι η βαθιά μηχανική μάθηση (deep learning). Ονομάστηκε deep learning οταν ο Geoffrey Hinton έδωσε τη φράση deep για να περιγράψει το δίκτυο πολλαπλών στρωμάτων (A Fast Learning Algorithm for Deep Belief Nets, 2006). Έκτοτε έχει χρησιμοποιηθεί εκτενώς και έχουν αναπτυχθεί διάφορα μοντέλα. Οι εφαρμογές των μοντέλων μηχανικής μάθησης συναντώνται σε διάφορους τομείς όπως στα self driving cars, speech recognition, κατηγοριοποίηση εικόνων κ.α. Η κατηγοριοποίηση εικόνων μάλιστα έχει χρησιμοποιηθεί αρκετά και στον τομέα της ιατρικής, όπου εκπαιδευμένα μηχανικά μοντέλα μπορούν να επιφέρουν ιατρική διάγνωση.

Το μεγάλο προτέρημα της βαθιάς μηχανικής μάθησης έναντι των άλλων μοντέλων, είναι ότι πρόκειται για ένα πολυεπίπεδο μοντέλο. Τα πολλαπλά διαδοχικά επίπεδα (multi - level), τα οποία συνδέονται το ένα με το άλλο βοηθούν να επιτευχθούν ακριβέστερες προβλέψεις σε περιπτώσεις όπου τα συμβατικά μοντέλα μηχανικής μάθησης υστερούσαν.

Στα συγκεκριμένα μοντέλα λοιπόν κάθε στοιχείο που διέρχεται στο σύστημα περνάει από όλα τα επίπεδα που έχει σχεδιαστεί το μοντέλο. Με την πολύ επίπεδη εκπαίδευση του μοντέλου, επιτυγχάνεται η εκπαίδευση του, χωρίς ωστόσο να απαιτείται τεράστιος όγκος δεδομένων για εκπαίδευση.

3.3.2. Οφέλη και Πλεονεκτήματα

Τα πλεονεκτήματα των μοντέλων της μηχανικής μάθησης έναντι των απλών μοντέλων μάθησης είναι αρκετά, και μάλιστα είναι ο λόγος της μεγάλης δημοφιλίας τους. Ένα από τα σημαντικότερα οφέλη είναι η ικανότητα των αλγορίθμων να πραγματοποιούν αυτόματα feature extraction από τα raw data. Στα deep learning μοντέλα δεν είναι δηλαδή απαραίτητο πριν την εκπαίδευση του μοντέλου να προηγηθεί κάποιος άλλος αλγόριθμο ο οποίος να προσδιορίζει τη σημαντικότητα των μεταβλητών. Ο ίδιος ο αλγόριθμος θα ανακαλύψει τις μεταβλητές που περιγράφουν καλύτερα το πρόβλημα, και θα χρησιμοποιήσει τα features που θα θεωρήσει ως πιο απαραίτητα για την πρόβλεψη.

Ως επακόλουθο αυτού, ένα ακόμη πλεονέκτημα των μοντέλων της βαθιάς μηχανικής μάθησης είναι η ικανότητα των μοντέλο στο χειρισμό των μεγάλων δεδομένων (big data). Ακόμη δηλαδή και αν πρόκειται να χρησιμοποιηθεί ένας μεγάλος όγκου δεδομένων για την εκπαίδευση του μοντέλου, αυτό δεν θα αντιμετωπίσει δυσκολίες στον χειρισμό των δεδομένων, αφού το μοντέλο θα χρησιμοποιήσει τις μεταβλητές που θα χρειάζονται περισσότερο (Deep Learning of Representations for Unsupervised and Transfer Learning, 2012).

3.4 Νευρωνικά Δίκτυα

3.4.1 Recurrent Neural Networks - RNN

Τα νευρωνικά δίκτυα είναι ένας κλάδος βαθιάς μηχανικής μάθησης που είναι εμπνευσμένος από τον τρόπο που λειτουργεί ο ανθρώπινος εγκέφαλος. Ο ανθρώπινος εγκέφαλος μπορεί να αντιληφθεί πολύ γρήγορα τι συμβαίνει γύρω του, χάρη σε μία πληθώρα νευρώνων οι οποίοι αντιδρούν μεταξύ τους πολύ γρήγορα. Όπως ακριβώς ένα ανθρώπινο νευρωνικό δίκτυο εκπαιδεύεται σιγά σιγά, και μαθαίνει πληροφορίες, είτε κρατώντας στη μνήμη του δεδομένα είτε ξεχνώντας τα έτσι ακριβώς χτίζεται και το τεχνητό νευρωνικό δίκτυο (Artificial Neural Network ANN).

Η εκπαίδευση των νευρωνικών δικτύων γίνεται τροφοδοτώντας το με πληροφορίες. Το ίδιο το μοντέλο αποφασίζει ποιες πληροφορίες χρειάζεται να θυμάται και ποιες πληροφορίες είναι προτιμότερο να ξεχάσει. Όταν πραγματοποιηθεί η εκπαίδευση του δικτύου τότε θα μπορεί το μοντέλο να χρησιμοποιηθεί σε προβλέψεις, δίνοντας του δεδομένα που δεν είχε ξαναδεί πριν.

Κάθε μοντέλο νευρωνικού δικτύου αποτελείται από εσωτερικούς βρόγχους, τις πύλες (Πύλη εισόδου - input, και πυλη εξόδου - output). Για κάθε πληροφορία που εισέρχεται σε ένα βρόχο υπάρχει και ένα κέντρο βάρους που συνδέει τους βρόγχους μεταξύ τους. Όταν εισέλθει το σύνολο πληροφοριών σε ένα νευρώνα τότε αυτός ενεργοποιείται, εφαρμόζοντας μία συνάρτηση (συνάρτηση ενεργοποίησης) για να προσδιορίσει την πληροφορία που θα

εξέλθει από τον νευρώνα, η οποία θα χρησιμοποιηθεί ως είσοδος στον επόμενο νευρώνα (Deep learning in spiking neural networks, 2019).

Ένα διαδεδομένο είδος βελτιωμένων νευρωνικών δικτύων είναι τα ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent Neural Networks - RNN). Η βελτίωση που προσφέρουν έγκειται στο γεγονός ότι σε κάθε νευρώνα στον οποίο ενεργοποιείται η συνάρτηση ενεργοποίησης, εκτός από το κέντρο βάρους τους, λαμβάνεται υπόψη και η πληροφορία που είχαν από την προηγούμενη ενεργοποίηση της συνάρτησης στον προηγούμενο νευρώνα, δημιουργώντας δηλαδή μία πύλη μνήμης (Forget gate). Το RNN μοντέλο χρησιμοποιούνται ευρέως στην πρόβλεψη των χρονοσειρών χάρη στην ικανότητα τους να μεταβιβάζουν πληροφορίες πίσω το δίκτυο και επομένως να θυμούνται τα προηγούμενα δεδομένα.

Πλέον υπάρχουν διάφοροι τρόποι όπου μπορεί να εκπαιδευτεί ένα μοντέλο νευρωνικού δικτύου. Ο πιο δημοφιλής τρόπος είναι ο backpropagation (οπίσθια ανατροφοδότηση). Το μοντέλο ξεκινάει με μία τυχαία επιλογή των τιμών του κέντρου βάρους. Οι τιμές των κέντρων βάρους δείχνουν πόσο έντονη και ισχυρή είναι η σύνδεση μεταξύ των νευρώνων. Στη συνέχεια χρησιμοποιεί εμπρόσθια ανατροφοδότηση (forward propagation) για να δημιουργήσει προβλέψεις. Με τον αλγόριθμό του backpropagation δημιουργείται ένα σφάλμα (απώλεια - loss). Χρησιμοποιώντας την οπίσθια ανατροφοδότηση (backpropagation) υπολογίζεται το σφάλμα στη συνάρτηση απώλειας. Αναλόγως με το ποσό του σφάλματος, αναπροσαρμόζονται οι συνδέσεις μεταξύ των νευρώνων.

Στην περίπτωση της εποπτευόμενης μάθησης (supervised learning) το μοντέλο εκπαιδεύεται συγκρίνοντας την προβλεπόμενη τιμή που παράγει το μοντέλο, και την τιμή στόχο (target) που του έχει δοθεί. Έτσι κατά την διάρκεια των ανατροφοδόσεων του μοντέλου, αυτό αναπροσαρμόζει το κέντρο βάρους των παραμέτρων, μέχρι να ελαχιστοποιηθεί η απώλεια, χρησιμοποιώντας μία σταθερά εκμάθησης (learning rate) για κάθε epoch. Το μοντέλο θα συνεχίσει να εκπαιδεύεται (και να αναπροσαρμόζει τα κέντρα βάρους) είτε μέχρι να ολοκληρωθεί ο κύκλος των run (epoch) είτε αν έχει επιτευχθεί η καλύτερη δυνατή μεταβολή της μείωσης του λάθος. Με την εκπαίδευση του μοντέλου δηλαδή επιτυγχάνεται η εύρεση των κέντρων βάρων που ελαχιστοποιούν το σφάλμα.

Έτσι τα RNN αποτελούνται από εσωτερικούς βρόγχους και μπορούν να καταλάβουν τη σημαντικότητα των δεδομένων για ένα πρόβλημα. Καθώς προχωράει η πληροφορία μέσα από τους βρόγχους το μοντέλο επιλέγει αν θα κρατήσει αυτήν την πληροφορία στη μνήμη του ή αν θα τη διαγράψει. Κάθε μοντέλο αποτελείται από k+2 επίπεδα, όπου το k είναι ο αριθμός των κρυμμένων επιπέδων και τα 2 είναι το επίπεδο εισόδου και εξόδου.

3.4.2 Συνάρτηση Softmax

Η συνάρτηση Softmax είναι από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στα νευρωνικά δίκτυα της μηχανικής μάθησης. Με τη χρήση της συνάρτησης softmax επιτυγχάνεται η κανονικοποίηση των δεδομένων (normalization). Είναι μία μαθηματική συνάρτηση με σκοπό τη μετατροπή ενός διανύσματος τιμών σε πιθανότητες, οι οποίες αθροίζουν στη μονάδα.

3.5 Long Short-Term Memory

3.5.1 Μοντέλο LSTM

Τα νευρωνικά δίκτυα LSTM (Long Short-Term Memory) είναι ένα είδος επαναλαμβανόμενων δικτύων με ανατροφοδότηση που ανήκουν στον τομέα του deep learning. Έχουν γίνει ευρέως δημοφιλή χάρη στην ικανότητα τους να συνδυάζουν πληροφορίες από το πολύ προγενέστερες συνδέσεις. Όσο μεγαλύτερο είναι τα χρονικό χάσμα μεταξύ των συνδέσεων που πρέπει να χρησιμοποιηθούν τόσο μη αποτελεσματικά γίνονται τα RNN μοντέλα.

Τα επαναλαμβανόμενα νευρωνικά δίκτυα LSTM διατηρούν στη μνήμη τους πληροφορίες των historical data για μεγάλες χρονικές περιόδους και σύμφωνα με αυτά μπορούν να κάνουν μακροπρόθεσμες προβλέψεις. Για πρώτη φορά εισήχθησαν από τους Hochreiter & Schmidhuber (1997) και έκτοτε έχουν βελτιωθεί και χρησιμοποιηθεί αρκετα.

Ενώ λοιπόν τα RNN (Recurrent neural networks) μοντέλα βασίζονται στο βραχυχρόνια μνήμη (short term memory) το μοντέλο LSTM έχει την ικανότητα να αποθηκεύει πληροφορίες των historical data. Η βελτίωση που προσφέρουν τα LSTM δίκτυα σε σχέση με τα RNN είναι ότι αποθηκεύοντας τα δεδομένα στην μακροχρόνια μνήμη καλύπτεται η αδυναμία που εμφανίζουν τα RNN όταν υπήρχε μεγάλο χρονικό διάστημα μεταξύ των γεγονότων.

3.5.2 Αρχιτεκτονική LSTM

Τα δίκτυα LSTM μοιάζουν στην αρχιτεκτονική με τα δίκτυα RNN. Αποτελούνται και αυτά από πολύ επίπεδα συνδεδεμένα μεταξύ τους, με το σχήμα των επιπέδων να είναι k+2. Έχουν σχεδιαστεί έχοντας επίσης τρεις πύλες, όπως ακριβώς και τα δίκτυα RNN, με τη διαφορά ότι αποτελούνται και από κελιά μνήμης. Με την χρήση των κελιών μνήμης το μοντέλο μπορεί να αποθηκεύσει αλλά και να ανακτήσει πληροφορίες ακόμη και αν έχει παρέλθει μεγάλο χρονικό διάστημα μεταξύ των συνδέσεων. Κατα τη διάρκεια της εκπαίδευσης ενός LSTM δικτύου, το μοντέλο επιλέγει ποιες πληροφορίες πρέπει να θυμάται ή να ξεχάσει αποθηκεύοντας πληροφορίες στα κελιά μνήμης, για όσο διάστημα η πύλη εισόδου (Input gate) παραμένει κλειστή.

Το μοντέλο αποτελείται από τρεις πύλες (gates), και τα κελιά μνήμης:

- Forget Gate
- Input Gate
- Cell
- Output Gate

Η επιλογή των πληροφοριών γίνεται στην πύλη μνήμης (forget gate) (Single Layer & Multi - layer Lon Short - Term Memory (LSTM) model with Intermediate variables for Weather Forecasting, 2018). Η κάθε πληροφορία που εισέρχεται στην πύλη μνήμης ελέγχεται για το αν πρέπει να καταγραφεί ή όχι στο κελί μνήμης. Στη συνέχεια ενεργοποιείται η σιγμοειδής

συνάρτηση (Sigmoid function) σύμφωνα με την οποία το μοντέλο αποφασίζει αν θα αποθηκεύσει ή όχι αυτήν την πληροφορία τους στα κελιά μνήμης. Με τη σιγμοειδής συνάρτησης λαμβάνονται τιμές της κλίμακας 0-1. Η τιμή που θα λάβει η σιγμοειδής συνάρτηση θα καθορίσει το ποσοστό της πληροφορίας που θα διατηρήσει τη μνήμη του. Όσο πιο όσο πιο κοντά είναι στο μηδέν, τόσο το μοντέλο τη θεωρεί μη σημαντική οπότε θα ¨ξεχάσει" αυτήν την πληροφορία, ενώ αντιθέτως όσο πιο κοντά η τιμή είναι κοντά στο 1, τόσο το μοντέλο θεωρεί σημαντική αυτήν την πληροφορία οπότε και θα τη "θυμάται" στο κελί μνήμης (Long Short-Term Memory based deep recurrent neural networks for Large scale acoustic modeling, 2014)

Στη συνέχεια επόμενο σημαντικό βήμα είναι να αποφασιστεί πιο μέρος της πληροφορίας θα μπορούσε να αποδεσμευτεί από το κελί μνήμης. Η διαδικασία αυτή λαμβάνει χώρα στην πύλη μνήμης (forget gate). Στην πύλη μνήμης θα εισέλθει η πληροφορία από το προηγούμενο στάδιο (πύλη εξόδου) καθώς και η τιμή (0-1) που έχει ληφθεί από τη σιγμοειδή συνάρτηση. Έτσι σε κάθε βήμα λαμβάνει την πληροφορία από το προηγούμενο βήμα, και δημιουργείται ένα διάνυσμα δεδομένων με τις υποψήφιες πληροφορίες, όπου και αξιολογείται ποιες πληροφορίες θα πρέπει να διατηρηθούν στη μνήμη ή όχι, και στη συνέχεια να τις αποθηκεύσει. (Hochreiter and Schmidhuber, 2012).

Υπάρχουν βέβαια αρκετές παραλλαγές του μοντέλου των LSTM, αφού λόγω της δημοτικότητας του είναι ευρέως χρησιμοποιούμενα. Κάθε μικρό διαφορά που μπορεί να τροποποιήσει το μοντέλο αποτελεί και μία παραλλαγή του κλασικού μοντέλου LSTM.

3.5.3 Παράμετροι στα LSTM

Hidden layers

Μία βασική παράμετρος που χρειάζεται είναι το πλήθος των κρυφών επιπέδων (hidden layers) που θα έχει το τελικό μοντέλο LSTM. Υπάρχουν κάποιοι βασικοί κανόνες σχετικά με την εύρεση της καλύτερης τιμής των κρυμμένων επιπέδων, αλλά ωστόσο αυτό δεν είναι απόλυτο. Αν τα δεδομένα είναι high - level, είναι προτιμότερο να χρησιμοποιούνται περισσότερα επίπεδα ώστε να επιτυγχάνεται η ορθότερη αφαίρεση των περιττών στοιχείων.

Ενώ όταν τα δεδομένα παρουσιάζουν γραμμικότητα θα μπορούσε να σχεδιαστεί ένα μοντέλο ακόμη και χωρίς κρυφά επίπεδα. Είναι σημαντικό να επισημανθεί ότι όσο περισσότερα επίπεδα έχει το μοντέλο, τόσο περισσότερο χρόνο αλλά και υπολογιστικών πόρων θα χρειαστεί για την εκπαίδευση του μοντέλου (Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture, 2012).

Learning rate

Ο ρυθμός μάθησης (learning rate) είναι η παράμετρος η οποία ελέγχει πόσο γρήγορα ή αργά το μοντέλο θα προσπαθήσει να εξηγήσει το πρόβλημα. Είναι σημαντικό να προσδιοριστεί η βέλτιστη τιμής της καθώς αυτή η παράμετρος θα προσδιορίζει το βαθμό απόκρισης του μοντέλου στην μεταβολή των μεταβολή των κέντρων βάρους, κάθε φορά που αυτά αλλάζουν. Όσο πιο μικρή είναι η τιμή τόσο πιο αργά αλλά σταθερά θα εκπαιδεύεται το μοντέλο. Ωστόσο υπάρχει ο κίνδυνος να είναι εξαιρετικά χρονοβόρα η διαδικασία της εκπαίδευσης, καθώς οι μικρές τιμές του ρυθμού μάθησης συχνά συνοδεύονται από πολλές epochs. Αντιθετα αν η τιμή είναι μεγάλη, θα επιτευχθεί καλύτερη ταχύτητα, αλλά ίσως μαθαίνει υπερβολικά γρήγορα, και προκληθεί αστάθεια λόγω των γρήγορων αλλαγών κατα την διαδικασία της εκπαίδευσης. Η τιμή του ρυθμού μάθησης είναι πάντα θετική, κλίμακας 0.0 - 1.0. Η πιο σύνηθες τιμή που χρησιμοποιείται είναι το 0.1, αλλά καθώς ο ρυθμός μάθησης αποτελεί πολύ σημαντική παράμετρο, συχνά παρατηρείται να δίνεται ένα διάνυσμα τιμών και συγκρίνοντας τα αποτελέσματα για κάθε τιμή, να αποφασίσει το μοντέλο ποια θα είναι η βέλτιστη τιμή.

Μάλιστα οι Ian Goodfellow, Yoshua Bengio, Aaron Courville αναφέρουν ότι ο προσδιορισμός της βέλτιστης τιμής του ρυθμού μάθησης είναι ίσως η σημαντικότερα παράμετρος. (Deep Learning (Adaptive Computation and Machine Learning series, 2016).

Early Stopping

Η τεχνική του early stopping βοηθάει στην ουσία το μοντέλο να προσδιορίσει το βέλτιστο αριθμό των epochs. Αν το μοντέλο έχει πολύ μικρό αριθμό epochs, έγκειται ο κίνδυνος να μην προλάβει να εκπαιδευτεί σωστά το μοντέλο. Ενώ δηλαδή δεν έχει φτάσει τη βέλτιστη τιμή μείωσης του σφάλματος, το μοντέλο ωστόσο θα σταματήσει να εκπαιδεύεται γιατί δεν θα έχει άλλη epoch να καλύψει. Το γεγονός αυτό οδηγεί σε underfit των δεδομένων. Με τον

όρο underfit εννοούμε ότι το μοντέλο δεν μπόρεσε ούτε στα δεδομένα που έλαβε για εκπαίδευση (training data) να προσαρμοστεί και να μάθει από την διαδικασία. Εντούτοις ένα μοντέλο που δεν προσαρμόστηκε σωστά στα training data δεν θα μπορέσει να δώσει σωστές και ακριβείς προβλέψεις. Στην αντίθετη πλευρά, για περιπτώσεις δηλαδή με πολύ μεγάλη τιμή των epochs, το μοντέλο κινδυνεύει από overfitting. Δίνοντας του πολλές epochs, θα μάθει τόσο καλά τα training data, που θα έχει μεγάλη ακρίβεια στην πρόβλεψη τους, και μικρό ποσοστό λάθος. Όταν όμως λάβει δεδομένα διαφορετικά από τα training data, τότε δεν θα είναι σε θέση να μπορέσει να κάνει πρόβλεψη. Με την χρήση του Early Stopping η εκπαίδευση του μοντέλου μπορεί να σταματήσει πριν ολοκληρωθούν όλες οι epochs, αλλά έχοντας πετύχει την καλύτερη δυνατή μεταβολή στο συνάρτηση απώλειας (loss).

Dropout

Το Dropout είναι μία μέθοδος regularization, όπου σκοπό έχει να προστατέψει το μοντέλο από overfitting. Με τη χρήση του Dropout επιλέγονται τυχαία κάποιοι νευρώνες οι οποίοι θα εξαιρεθούν από την διαδικασία μάθησης, δηλαδή δεν ανανεώνονται τα κέντρα βάρους τους. Το dropout παίρνει τιμές κλίμακας 0-1, καθώς εκφράζει την πιθανότητα (ποσοστό) να απενεργοποιηθεί ένας νευρώνας. (A Theoretically Grounded Application of Dropout in Recurrent Neural Networks, 2016).

4. Μεθοδολογία

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται λεπτομερώς η μεθοδολογία που ακολουθήθηκε. Αρχικά παρουσιάζεται ο τρόπους που πραγματοποιηθεί η συλλογή των δεδομένων. Επιλέχθηκαν οι μεταβλητές που παρουσιάζουν ενδιαφέρον, καθώς θεωρήθηκαν ότι μπορούν να περιγράψουν τα ποιοτικά χαρακτηριστικά των κατοίκων των περιοχών. Οπότε έγινε η υπόθεση ότι βάσει των συγκεκριμένων χαρακτηριστικών θα μπορούσε να εξηγηθεί αλλά και να προβλεφθεί οι τιμές των σπιτιών. Στη συνέχεια αναλύεται ο τρόπος που έγινε η επεξεργασία των δεδομένων. Παρουσιάζεται αναλυτικά η ενοποίηση τους και η δημιουργία

του τελικού dataset που θα χρησιμοποιηθεί στα προβλεπτικά μοντέλα. Εξηγείται ακόμη ο τρόπος χειρισμού των δεδομένων σε περίπτωση που εντοπίζονται ελλιπείς τιμές (missing data). Αφού πραγματοποιηθεί το πρώτο στάδιο επεξεργασίας των δεδομένων στη συνέχεια γίνεται μία περιγραφική ανάλυση αυτών. Πριν προηγηθεί ο σχεδιασμός των μοντέλων είναι σημαντικό να έχουν κατανοηθεί οι μεταβλητές, και να αναφερθούν τα συμπεράσματα που μπορεί να προκύψουν από την επεξήγηση τους. Αφού λοιπόν ολοκληρωθεί και η ανάλυση των μεταβλητών, στη συνέχεια ακολουθεί ο σχεδιασμός των προβλεπτικών μοντέλων μηχανικής μάθησης. Παρουσιάζεται αναλυτικά ο τρόπου δημιουργίας του LSTM μοντέλου, καθώς και η επεξήγηση των παραμέτρων που επιλέχθηκαν. Στην συνέχεια ακολουθεί ο αντίστοιχος σχεδιασμός αλλά και η επεξήγηση των παραμέτρων για το XGBoost μοντέλο. Μετά των σχεδιασμό, γίνεται η εκπαίδευση των δύο μοντέλων χρησιμοποιώντας ένα υποσύνολο των τελικών δεδομένων (train data). Τελικό στάδιο πριν την ολοκλήρωση είναι η αξιολόγηση των μοντέλων. Πραγματοποιείται για κάθε μοντέλο η εκτίμηση που θα προέβλεπε, βασιζόμενο στο άλλο υποσύνολο των δεδομένων (test data). Η εκτίμηση αυτή συγκρίνεται με την πραγματική τιμή που θα έπρεπε να προέβλεπε το μοντέλο. Έτσι συγκρίνοντας τα δύο αυτά στοιχεία, αλλά και με την χρήση διάφορων αριθμητικών και οπτικών δεικτών, μπορεί να γίνει η αξιολόγηση τους. Αναλύονται τα αποτελέσματα του κάθε μοντέλου και γίνεται σύγκριση των αποτελεσμάτων, σε σχέση με την η ικανότητα τους να εκτιμήσουν την ταση της τιμής των σπιτιών.

4.2 Περιγραφή Μεθοδολογίας

Αρχικά έγινε η περιήγηση και κατανόηση των δεδομένων που προσφέρονται από την ιστοσελίδα ανοιχτών δεδομένων της Σκωτίας http://statistics.gov.scot. Στη συνέχεια έγινε η επιλογή των μεταβλητών που θα χρησιμοποιηθούν ως ανεξάρτητες μεταβλητές στο μοντέλο πρόβλεψης καθώς θεωρείται ότι μπορούν να δώσουν τα ποιοτικά χαρακτηριστικά των μικρών γεωγραφικών περιοχών της Σκωτίας.

Μετά τη επιλογή των δεδομένων ακολούθησε η δημιουργία των dataset στην ιστοσελίδα, όπου για κάθε ανεξάρτητη μεταβλητή δινόταν η διάσταση του χώρου (2011 Data Zone) και του χρόνου. Όταν δημιουργήθηκαν όλα τα επιθυμητά dataset έγινε η συλλογή των

δεδομένων. Η επεξεργασία, των δεδομένων, η ανάλυση τους, καθώς και ο σχεδιασμός του μοντέλου μηχανικής μάθησης έγιναν με τη χρήση της γλώσσας προγραμματισμού Python και με την βιβλιοθήκη keras.

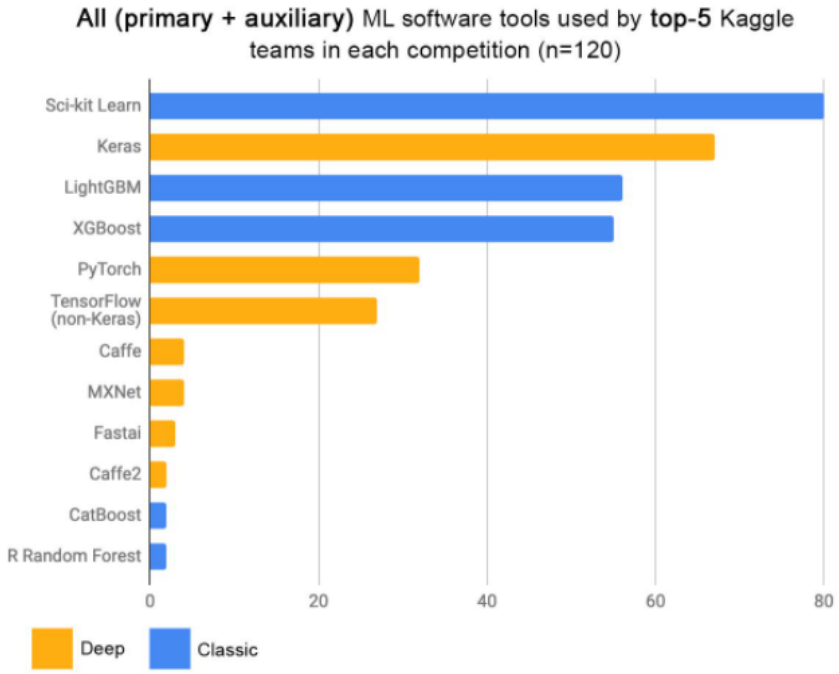
Η γλώσσα προγραμματισμού Python δημιουργήθηκε από τον Guido van Rossum ως μία διάδοχος της ABC. Η πρώτη εκδοχή δημοσιεύθηκε το 1991 και περιλάμβανε την Python 0.9.0, ενώ σχεδόν μία δεκαετία αργότερα, το 2000 κυκλοφόρησε η έκδοση 2.0. Το 2008 ανακοινώθηκε η 3.0, ενώ πλέον η τελευταία έκδοση που κυκλοφορεί είναι η 3.10.4 (https://www.python.org/).

Πρόκειται μία ερμηνευμένη αντικειμενοστραφής γλώσσα προγραμματισμού η οποία μάλιστα γίνεται ολοένα και πιο δημοφιλής. Είναι γεγονός ότι παρατηρείται μία συνεχής αύξηση των ατόμων που την μαθαίνουν και την χρησιμοποιούν. Μάλιστα είναι εντυπωσιακό ότι μέχρι στιγμής δεν έχουν παρατηρηθεί ακόμη πτωτικές τάσεις στον αριθμό των χρηστών. Ανήκει στις γλώσσες υψηλού επιπέδου, η οποία όμως έχει πολύ σαφής σύνταξη Η Python χρησιμοποιείται κυρίως ως το frontend των βιβλιοθηκών της μηχανικής μάθησης. Από τις πιο συχνά χρησιμοποιούμενες βιβλιοθήκες είναι το TensorFlow και το PyTorch. (Q. Zhang, L. Xu, X. Zhang, B. Xu, 2021)

Η βιβλιοθήκη PyTorch είναι μία βιβλιοθήκη ανοιχτού κώδικα (open source) που υποστηρίζει τη μηχανική μάθησης αλλά και τη βαθιά μάθηση. Έχει σχεδιαστεί από το Facebook και καθώς είναι Python-based είναι εξαιρετικά εύκολη για χρήση και ανάπτυξή της (N. Ketkar, J. Moolayil, 2021)

Το TensorFlow πρόκειται στην ουσία για μία ανοιχτού κώδικα πλατφόρμα μηχανικής μάθησης. Στα βασικά του πλεονεκτήματα ανήκει η ικανότητα εκτέλεσης λειτουργιών με χαμηλό επίπεδο CPU και GPU. Επίσης οι εκτελέσεις των λειτουργιών μπορούν να συμβούν σε πολλές συσκευές, δημιουργώντας έτσι συμπλέγματα GPU. Για τους σκοπούς της συγκεκριμένης εργασίας θα χρησιμοποιηθούν οι βιβλιοθήκες του Keras.(L.Parisi, R, Ma, N, RaviChandran, M. Lanzillotta, 2021)

Το Keras είναι ένα API της πλατφόρμας του TensorFlow γραμμένο σε Python, και το παρέχει αλγορίθμους της βαθιάς μηχανικής μάθησης. Πολλές μεγάλες εταιρίες χρησιμοποιούν το Keras για να χτίσουν τα μοντέλα τους, όπως για παράδειγμα το Netflix και η Uber.Μάλιστα είναι από τα πιο συχνά χρησιμοποιούμενα tools στο Kaggle, και συγκεκριμένα έρχεται το πρώτο πιο δημοφιλή στα deep learning. (https://keras.io/)



Εικόνα 2: Διάγραμμα των πιο συχνά χρησιμοποιούμενων software tools (Πηγή: keras.io)

5. Μέθοδος Περίπτωσης

5.1 Ανάλυση Προβλήματος

Το πρόβλημα που εξετάζεται στα πλαίσια της εργασίας είναι η πρόβλεψη της κίνησης των σπιτιών σύμφωνα με τη γεωγραφική περιοχή που ανήκουν (Data Zones 2011). Ωστόσο σκοπός δεν είναι απλώς η έρευσης της συνεχούς μεταβλητής (μία αξιακή τιμή δηλαδή). Αντιθέτως στόχος του μοντέλου πρόβλεψης θα είναι η εύρεση της τάσεις των τιμών. Εάν δηλαδή θα υπάρξει άνοδος στις τιμές των σπιτιών σε σχέση με την προηγούμενη περίοδο αναφοράς (έτος) ή αν θα υπάρξει μείωση των τιμών. Οπότε κινούμενοι προς αυτήν την

κατεύθυνση το μοντέλο θα προσπαθήσει να δώσει πρόβλεψη για την μεταβλητή της τάσης, η οποία θα είναι μία κατηγορική μεταβλητή με δύο κατηγορίες, την αύξηση ή την μείωση των τιμών. Έτσι η προσέγγιση του προβλήματος θα γίνει θεωρώντας το ως classification πρόβλημα. Σε αυτήν την περίπτωση η εξαρτημένη μεταβλητή θα μπορεί να πάρει μόνο συγκεκριμένες τιμές (κλάσεις). Για να μετατρέψουμε το πρόβλημα σε classification, υπολογίζουμε τη μεταβολή της τιμής των σπιτιών μεταξύ τους έτους (N) και του προηγούμενου έτους (N-1) (Τιμή_έτος_N - Τιμή_έτος_N-1). Έτσι κατηγοριοποιούμε σε δύο περιπτώσεις, ανάλογα με την τιμή της ανεξάρτητης μεταβλητής (y):

- y=1, όταν η ανεξάρτητη μεταβλητής (y) είναι ίση με την μονάδα τότε θεωρούμε ότι οι τιμές των σπιτιών παρουσιάζουν αύξηση ή είναι στα ίδια επίπεδα σε σχέση με την προηγούμενη χρονιά (Τιμή_έτος_N - Τιμή_έτος_N-1 >=0)

- y= 0, αντίθετα όταν η ανεξάρτητη μεταβλητής (y) είναι ίση με το μηδέν τότε θεωρούμε ότι οι τιμές των σπιτιών παρουσιάζουν μείωση με την προηγούμενη χρονιά (Τιμή_έτος_N - Τιμή_έτος_N-1 < 0)

```
y_list = list()

for i in range(data_final.shape[0]):
    if math.isnan(data_final['Price-1'][i]):
        y=np.nan
    else:
        y=1 if data_final['Price'][i]-data_final['Price-1'][i]>=0 else 0
    y_list.append(y)

data_final['y']=y_list
```

Εικόνα 3: Δημιουργία της μεταβλητής κλάσης y

5.2 Συλλογή και επεξεργασία δεδομένων

Το πρώτο μας βήμα είναι να εισάγουμε τις απαραίτητες βιβλιοθήκες.

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
from scipy import stats
import random
import re
import math
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers.core import Dropout
from keras.layers import Dense
from sklearn.metrics import mean_squared_error
from sklearn.metrics import explained_variance_score
from sklearn.metrics import r2_score
```

Εικόνα 4: Εισαγωγή βιβλιοθηκών

Οι πρώτες δύο βιβλιοθήκες είναι οι βασικές βιβλιοθήκες επεξεργασίας DataFrame δεδομένων στην python. Στη συνέχεια εισάγουμε κάποιες βιβλιοθήκες που περιέχουν μαθηματικές φόρμουλες, καθώς και τις βιβλιοθήκες sklearn για την αξιολόγηση των αποτελεσμάτων του μοντέλου πρόβλεψης. Εισάγουμε την matplotlib βιβλιοθήκη για την οπτικοποίηση των δεδομένων και τέλος την βιβλιοθήκη Keras με την χρήση της οποίας επιτεύχθηκε ο σχεδιασμός του μοντέλου μηχανικής μάθησης. Επόμενο βήμα είναι να καλέσουμε τα δεδομένα. Κάθε dataset περιλαμβάνει τις ανεξάρτητες μεταβλητές που θα χρησιμοποιήσουμε στο μοντέλο μας. Απαραίτητη μεταβλητή είναι επίσης η γεωγραφική θέση και το έτος αναφοράς.

Στην ιστοσελίδα ανοιχτών δεδομένων της Σκωτίας (http://statistics.gov.scot) επιλέγουμε τις μεταβλητές που θεωρούμε ότι είναι χρήσιμα χαρακτηριστικά της ποιότητας ζωής των κατοίκων μίας περιοχής και κατ' επέκταση επηρεάζουν τις τιμές των σπιτιών των αντίστοιχων περιοχών.

Οι μεταβλητές που επιλέχθηκαν είναι οι παρακάτω:

- Ο Αριθμός Γεννήσεων
- Ο αριθμός θανάτων
- Ο πληθυσμός * (για τον υπολογισμό του ποσοστού γεννήσεων και θανάτων)
- Το μισθολογικό χάσμα μεταξύ των δύο φύλων σε πλήρης και σε μερική απασχόληση
- Το ποσοστό ανδρών μεταξύ 16-64 ετών με χαμηλά ή και καθόλου προσόντα
- Το ποσοστό γυναικών μεταξύ 16-64 ετών με χαμηλά ή και καθόλου προσόντα
- ποσοστό των ανθρώπων που εγκαταλείπουν το σχολείο
- Το ποσοστό των γυναικών που κάπνιζαν πριν τη γέννα
- Και το ποσοστό των ατυχημάτων φωτιάς

Δημιουργούμε τις συναρτήσεις correct_dataset και missing_data όπου τις εφαρμόζουν στα παραπανω datasets.

```
#function to stack the dataframe
def correct_dataset(df,start_year,feature,geo='Zone'):
    df.loc[:,'Feature Identifier']=df.loc[:,'Feature Identifier'].apply(lambda x: x.replace('http://statistics.gov.scot/id/statistical-geo
    n_columns = len(df.columns)
    name_columns = ['Code_'+geo,'Name_'+geo]
    s = int(start_year)-1
    for i in df.columns:
        if 'Feature' not in i:
            s = s+1
            str_year = str(s)
            name_columns.append(str_year)
    df.columns = name_columns
    df = df.set_index(['Code_'+geo,'Name_'+geo],append=True).stack(dropna=False).reset_index().drop('level_0',1)
    correct_name_cols = ['Code_'+geo,'Name_'+geo,'Year']
    correct_name_cols.append(feature)
    df.columns = correct_name_cols
    return df
```

Εικόνα 5: Δημιουργία συνάρτησης μετονομίας στηλών και crosstab του table

```python
#handling missing data, with interpolatelinear
#set year, as date, and then set indexes
def missing_data(df,geo='Zone'):
    col =df.columns.drop(['Code_'+geo,'Name_'+geo,'Year'])
    #col.append(colname)
    df.loc[:,'Year'] = pd.to_datetime(df.loc[:,'Year'], format='%Y')
    df = df.set_index('Year')
    df_interpol = df.groupby(['Code_'+geo,'Name_'+geo])\
                .resample('Y')\
                .mean()
    df_interpol[col]=df_interpol[col].interpolate()
    df_interpol = df_interpol.reset_index()
    df_interpol.loc[:,'Year'] = df_interpol.loc[:,'Year'].dt.year.astype(str)
    return df_interpol
```

Εικόνα 6: Δημιουργία συνάρτησης χειρισμού των ελλιπών τιμών

Με συνάρτηση correct_dataset ονομάζουμε τις στήλες που αναφέρονται στη γεωγραφική περιοχή ως Code_Zone και Name_Zone και στη συνέχεια γίνεται ένα crosstab του dataset ως προς τις μεταβλητές που δηλώνουν τον χρόνο (Year) και τις μεταβλητές της γεωγραφικής περιοχής (Code_Zone και Name_Zone)

Παράδειγμα μετασχηματισμού:

Αρχικό dataset:

0]:

	Feature Identifier	Feature Name	Population Estimates Summary (Current Geographic Boundaries): Age = All; Reference Period = 2014; Sex = All; measure type = Count	Population Estimates Summary (Current Geographic Boundaries): Age = All; Reference Period = 2015; Sex = All; measure type = Count	Population Estimates Summary (Current Geographic Boundaries): Age = All; Reference Period = 2016; Sex = All; measure type = Count	Population Estimates Summary (Current Geographic Boundaries): Age = All; Reference Period = 2017; Sex = All; measure type = Count
0	http://statistics.gov.scot/id/statistical-geog...	Culter - 01	898	901	897	894
1	http://statistics.gov.scot/id/statistical-geog...	Culter - 02	837	818	807	793
2	http://statistics.gov.scot/id/statistical-geog...	Culter - 03	696	669	640	624
3	http://statistics.gov.scot/id/statistical-geog...	Culter - 04	563	553	555	537
4	http://statistics.gov.scot/id/statistical-geog...	Culter - 05	684	694	671	663

Εικόνα 7: Απεικόνιση του αρχικού dataset

Dataset μετά την εφαρμογή του correct_dataset function:

	Code_Zone	Name_Zone	Year	Polulation
0	S01006506	Culter - 01	2014	898
1	S01006506	Culter - 01	2015	901
2	S01006506	Culter - 01	2016	897
3	S01006506	Culter - 01	2017	894
4	S01006506	Culter - 01	2018	850

Εικόνα 8: Απεικόνιση του dataset μετά την correct dataset function

Με τη συνάρτηση missing_data, χειριζόμαστε τις τιμές που λείπουν. Όπως αναφέρουν και οι Max kuhn and Kjell Jonson (Applied Predictive Modeling 1st ed. 2013), μπορούμε να εκτιμήσουμε τα δεδομένα που λείπουν χρησιμοποιώντας πληροφορίες από τις υπόλοιπες τιμές. Χρησιμοποιήθηκε η μέθοδος του Linear Interpolation, με την οποία η εκτίμηση μίας τιμής γίνεται γραμμικά, λαμβάνοντας υπόψη τα δύο γειτονικά σημεία. Δεδομένου ότι έχουμε ιστορικά δεδομένα, θεωρούμε ότι όταν λείπει μία τιμή για μία χρονιά σε μία συγκεκριμένη γεωγραφική περιοχή, τα δύο γειτονικά σημεία της μπορούν να την προσδιορίσουν.

Στη συνέχεια γίνεται η εισαγωγή των δεδομένων. Τα δεδομένα όπως έχει ήδη αναφερθεί έχουν συλλεχθεί από το site OGD της Σκωτίας. Για τη δημιουργία των dataset, επιλέχθηκαν οι επιθυμητές μεταβλητές για κάθε έτος, και στη συνέχεια η γεωγραφική περιοχή. Τα έτη τα οποία θα χρησιμοποιήσουμε είναι από το 2014 εώς το 2018, και η γεωγραφική περιοχή θα είναι το 2011 Data Zone.

Το πρώτο dataset αφορά τον αριθμό γεννήσεων και θανάτων, με διαχωρισμό ώς προς το φύλλο (άνδρες, γυναίκες), ανά έτος και γεωγραφική περιοχή. Κάθως το συγκεκριμένο dataset δεν είχε μεταβλητές για το έτος 2018, θα το υπολογίσουμε ως τον μέσο όρο των προηγούμενων ετών.

```
births_deths['Deaths_female_2018'] = births_deths.iloc[:, [12,13,14,15]].mean(axis=1)
births_deths['Deaths_male_2018'] = births_deths.iloc[:, [16,17,18,19]].mean(axis=1)
```

Εικόνα 9: Υπολογισμός Μέσου όρου

Στην συνέχεια θα εφαρμόσουμε τις δύο συναρτήσεις, correct_dataset, και missing_values. Επιπλέον το δεύτερο βοηθητικό dataset που καλούμε, είναι αυτό του πληθυσμού (για κάθε έτος και σε κάθε γεωγραφική περιοχή) ώστε να υπολογίσουμε το ποσοστό των γεννήσεων και των θανάτων. Ενώνουμε τα δύο dataset κάνοντας merge βάσει της χρονιάς και της γεωγραφικής περιοχής, και στη συνέχεια υπολογίσουμε το ποσοστό γεννήσεων και θανάτων.

```
population = pd.read_csv('https://statistics.gov.scot/carts/download/1bafc10a-098c-4f03-b0da-edcb1fa0c0e9?format=csv')
population= correct_dataset(population,2014,'Polulation')
df_all = df_all.merge(population,how='left',on=['Code_Zone','Year','Name_Zone'])


#calculate the ratio for each columns
for j in range(3,7):
    new_col = df_all.columns[j]+'_ratio'
    df_all[new_col]=np.nan
    c=5+j
    for i in range(df_all.shape[0]):
        if df_all.iloc[i,7]!=0:
            df_all.iloc[i,c]=round((float(df_all.iloc[i,j])/float(df_all.iloc[i,7]))*1000,2)

df_all.head()
```

Εικόνα 10: Ενοποίηση των datasets

Το επόμενο dataset που δημιουργούμε εξετάζει τα ποιοτικά χαρακτηριστικά του πληθυσμού, όπως το αν έχουν ή όχι καθόλου εργασιακά προσόντα και ικανότητες, το αν έχουν ολοκληρώσει το σχολείο, ή το αν οι γυναίκες καπνίζουν πριν τη γέννα. Αντίστοιχα εφαρμόζουν και πάλι τη συνάρτηση correct_dataset για να το φέρουμε στην επιθυμητή μορφή. Όλα τα δεδομένα τα συγκεντρώνουμε σε ένα dataset (dfs_zone) και εφαρμόζουμε την συνάρτηση missing_data. Όπως φαίνεται και στην παρακάτω εικόνα (*Εικόνα 10*) αρχικά δημιουργούμε δύο κενές λίστες (list_data_with_null και data_ready) για να εξετάσουμε σε ποιο dataset πρέπει να εφαρμοστεί η συνάρτηση missing_data. Αν σε κάποιο dataset λείπει κάποια τιμή θα προστεθεί στη λίστα list_data_with_null διαφορετικά θα προστεθεί στην λίστα data_raedy. Μετα το πρώτο for loop και αφού δημιουργούνται οι δύο λίστες προκύπτει ότι η λίστα data_ready είναι κενή. Αυτό σημαίνει ότι σε όλα τα dataset υπάρχει έστω και μία τιμή που λείπει.

Οπότε εφαρμόζουμε σε όλα τα datasets την συνάρτηση missing_data, και συνέχεια δημιουργούμε ένα ενιαίο dataset (dfs_zone). Το πρώτο dataset που θα περάσει στο δεύτερο for loop θα δημιουργήσει τις πρώτες στήλες του dfs_zone, ενώ για όλα τα επόμενα γίνεται merge με το dfs_zone σύμφωνα με τη γεωγραφική περιοχή (*μεταβλητές: Cone_Zone και Name_Zone*) και το έτος (*μεταβλητή: Year*). Το dataset που δημιουργήθηκε περιλαμβάνει τις ανεξάρτητες μεταβλητές, το έτος και τη γεωγραφική θέση *(Εικόνα 11)*.

```
list_data_with_null=[]
data_ready=[]
for data in list_df_zones:
    if (data.isnull().any()).any():
         list_data_with_null.append(data)
    else:
        data_ready.append(data)
    print(data.isnull().sum())

#the data_ready list is empty. so we use only the list_data_with_null
#apply missing_data function
z=0

for i in range(len(list_data_with_null)):
    data = list_data_with_null[i].copy()
    df = missing_data(data)
    z=z+1
    if z==1:
        dfs_zone=df
    else:
        dfs_zone = dfs_zone.merge(df,how='left',on=['Code_Zone','Year','Name_Zone'])

dfs_zone.head()
```

Εικόνα 11: Εφαρμογή της συνάρτησης missing data

```
dfs_zone.head()
```

	Code_Zone	Name_Zone	Year	Births_F_ratio	Births_M_ratio	Deaths_F_ratio	Deaths_M_ratio	Educational attainment of school leavers	Ante-Natal Former Smoker	Ante-Natal Never Smoked	Ante-Natal Not Known	Fire Not Accidental
0	S01006506	Culter - 01	2014	6.68	5.57	0.00	4.45	6.10000	3.23	80.65	0.0	0.0
1	S01006506	Culter - 01	2015	6.66	5.55	3.33	2.22	6.23000	2.70	83.78	0.0	0.0
2	S01006506	Culter - 01	2016	5.57	6.69	3.34	5.57	6.05000	5.88	82.35	0.0	0.0
3	S01006506	Culter - 01	2017	5.59	2.24	4.47	2.24	5.88000	10.00	83.33	0.0	0.0
4	S01006506	Culter - 01	2018	1.18	5.88	2.94	3.82	5.63158	9.09	81.82	0.0	0.0

Εικόνα 12: Απεικόνιση του dataset dfs_zone

Το τρίτο dataset περιλαμβάνει τις επόμενες μεταβλητές που εξετάζουμε οι οποίες όμως έχουν κατηγοριοποιηθεί με την γεωγραφική περιοχή Council areas. Εφαρμόζουμε όπως και πριν την συνάρτηση correct_dataset για να το φέρουμε στην σωστή μορφή, στη συνέχεια

δημιουργούμε με την ίδιο τρόπο τις λίστες list_data_with_null και data_ready, και τέλος δημιουργούμε και πάλι την for loop για την εφαρμογή της συνάρτησης missing_data, όπου και προκύπτει το dfs_area dataset το οποίο περιλαμβάνει τις υπόλοιπες ανεξάρτητες μεταβλητές, το έτος αναφοράς και τη γεωγραφική θέση αναφοράς, δηλαδή την Council area (μεταβλητές: Code_Area και Name_Area).

Το τελευταίο dataset που καλούμε περιλαμβάνει τις τιμές των σπιτιών για τις περιοχές του 2011 Data Zone για τα έτη αναφοράς. Για τον λόγο αυτό καλούμε ένα βοηθητικό dataset που δείχνει την αντιστοιχία μεταξύ της γεωγραφικής περιοχής 2011 Data Zone και Council area (dataset: DZ_CA). Έτσι φέρνουμε τις στήλες που δείχνουν την περιοχή των Council areas στο dataframe prices. Στη συνέχεια εφαρμόζουμε τη συνάρτηση stack() για να γίνει transpose στον πίνακα και να μετατρέψουμε τις στήλες που δείχνουν την χρονιά σε γραμμές. Θέτουμε τις μεταβλητές της περιοχής (Cone_Zone, Name_Zone, LA_Code, LA_Name) ως index, για να μην συμπεριληφθούν στο transpose, και στη συνέχεια τις επαναφέρουμε ως στήλες και πάλι με το reset_index(). Έτσι φέρνουμε το dataframe prices στην επιθυμητή μορφή dfs_prices. (Εικονα 13)

```
prices = prices.merge(DZ_CA,how='left',left_on='Code_Zone',right_on='DZ2011_Code')
prices=prices.drop(['DZ2011_Code','DZ2011_Name'],axis=1)

df_prices = prices.set_index(['Code_Zone','Name_Zone','LA_Code','LA_Name'],append=True).stack().reset_index().rename(
columns={'level_5':'Year',0:'Price'}).drop('level_0',1)

data = df_prices.merge(dfs_zone,how='left',on=['Code_Zone','Year','Name_Zone'])
data = data.merge(dfs_area,how='left',left_on=['LA_Code','Year'],right_on=['Code_Area','Year'])
data=data.drop(['Code_Area','Name_Area'],axis=1)


del df_prices, dfs_zone, dfs_area
```

Εικόνα 13: Τρόπος χειρισμού της πληροφορίας των τιμών

```
prices.head()
```

	Code_Zone	Name_Zone	2014	2015	2016	2017	2018	LA_Code	LA_Name
0	S01006506	Culter - 01	192917.0	215003.0	198733.0	149727.0	122056.0	S12000033	Aberdeen City
1	S01006507	Culter - 02	381409.0	284539.0	161882.0	182182.0	177938.0	S12000033	Aberdeen City
2	S01006508	Culter - 03	205377.0	185767.0	168696.0	158000.0	147773.0	S12000033	Aberdeen City
3	S01006509	Culter - 04	223242.0	178700.0	131625.0	152019.0	138000.0	S12000033	Aberdeen City
4	S01006510	Culter - 05	186282.0	195236.0	173150.0	232350.0	NaN	S12000033	Aberdeen City

```
df_prices.head()
```

	Code_Zone	Name_Zone	LA_Code	LA_Name	Year	Price
0	S01006506	Culter - 01	S12000033	Aberdeen City	2014	192917.0
1	S01006506	Culter - 01	S12000033	Aberdeen City	2015	215003.0
2	S01006506	Culter - 01	S12000033	Aberdeen City	2016	198733.0
3	S01006506	Culter - 01	S12000033	Aberdeen City	2017	149727.0
4	S01006506	Culter - 01	S12000033	Aberdeen City	2018	122056.0

Εικόνα 14: Απεικόνιση του dataset df_price

Τέλος συγχωνεύουμε τα τρία dataframe (dfs_zone, dfs_areas και dfs_prices) σε ένα τελικό dataframe (data).

```
data.head()
```

	Code_Zone	Name_Zone	LA_Code	LA_Name	Year	Price	Births_F_ratio	Births_M_ratio	Deaths_F_ratio	Deaths_M_ratio	Educational attainment of school leavers	Ante-Natal Former Smoker	Ante-Natal Never Smoked	Ante-Natal Not Known	Fire Not Accidental	Gender PayGap full_time	Gender PayGap part_time	Female Adults with No qualifications	M qua
0	S01006506	Culter - 01	S12000033	Aberdeen City	2014	192917.0	6.68	5.57	0.00	4.45	6.10000	3.23	80.65	0.0	0.0	13.2	-26.0	8.1	
1	S01006506	Culter - 01	S12000033	Aberdeen City	2015	215003.0	6.66	5.55	3.33	2.22	6.23000	2.70	83.78	0.0	0.0	14.8	-1.9	10.4	
2	S01006506	Culter - 01	S12000033	Aberdeen City	2016	198733.0	5.57	6.69	3.34	5.57	6.05000	5.88	82.35	0.0	0.0	9.3	-9.8	8.9	
3	S01006506	Culter - 01	S12000033	Aberdeen City	2017	149727.0	5.59	2.24	4.47	2.24	5.88000	10.00	83.33	0.0	0.0	5.8	-13.3	11.8	
4	S01006506	Culter - 01	S12000033	Aberdeen City	2018	122056.0	1.18	5.88	2.94	3.82	5.63158	9.09	81.82	0.0	0.0	12.7	-20.0	8.2	

Εικόνα 15: Απεικόνιση της τελικής μορφής του dataset

5.3 Περιγραφή των δεδομένων

Αρχικά για να αποκτήσουμε μία πρώτη εικόνα των δεδομένων, εφαρμόζουμε το describe στο τελικό dataframe που έχουμε δημιουργήσει (data). Μπορούμε εύκολα να διακρίνουμε ότι το ποσοστό γεννήσεων και θανάτου κυμαίνονται κατά μέσο όρο στα ίδια επίπεδα, με τους θανάτους να υπερέχουν σχετικά. Επίσης παρατηρείται μεγάλο εύρος στην μεταβλητή Gender PayGap, η οποία εκφράζεται από κλάσμα:

- (Εισόδημα_ανδρών - Εισόδημα_γυναικών) / Εισόδημα_ανδρών.

```
data.describe()
```

	Price	Births_F_ratio	Births_M_ratio	Deaths_F_ratio	Deaths_M_ratio	Educational attainment of school leavers	Ante-Natal Former Smoker	Ante-Natal Never Smoked	Ante-Natal Not Known	Fire Not Accidental	Gender PayGap full_time	Gender PayGap part_time	Female Adults with No qualifications	Male Adults with No qualifications
count	3.054000e+04	30540.000000	30540.000000	30540.000000	30540.000000	30540.000000	30540.000000	30540.000000	30540.000000	30540.000000	30540.000000	30352.000000	30540.000000	30292.000000
mean	1.653171e+05	4.694899	4.984040	5.412272	5.133809	5.557110	12.271917	70.958779	3.098382	246.839558	7.468646	-9.015628	11.431260	11.642114
std	2.386696e+05	3.057859	3.182687	4.839428	3.511534	0.507289	9.632167	16.106502	5.660034	508.272135	8.900950	9.664472	3.477751	3.470009
min	1.673700e+04	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000	0.000000	0.000000	0.000000	-47.500000	-36.900000	5.600000	6.400000
25%	1.019585e+05	2.530000	2.730000	2.260000	2.660000	5.230000	5.260000	60.000000	0.000000	0.000000	4.000000	-15.200000	8.500000	9.000000
50%	1.442165e+05	4.300000	4.570000	4.270000	4.600000	5.595240	10.710000	72.220000	0.000000	109.200000	9.200000	-9.500000	11.000000	11.400000
75%	2.011342e+05	6.400000	6.790000	7.040000	6.920000	5.920000	17.650000	83.330000	4.760000	286.911000	12.700000	-3.100000	13.900000	13.900000
max	3.880480e+07	72.380000	70.480000	54.900000	39.970000	6.909090	100.000000	100.000000	80.000000	11607.100000	33.500000	24.642857	19.100000	20.800000

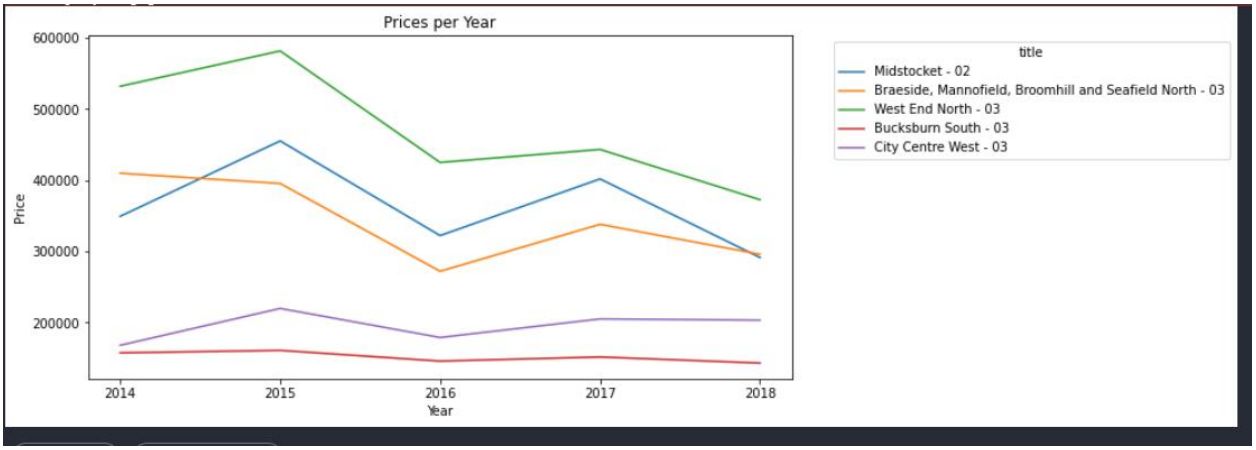
Εικόνα 16: Περιγραφική απεικόνιση των δεδομένων

Έπειτα αφού θεωρούμε ότι τα ποιοτικά χαρακτηριστικά μιας περιοχής επηρεάζουν την μεταβολή των τιμών των σπιτιών, δημιουργήσαμε δύο διαγράμματα για δύο διαφορετικές Council areas. Επιλέχθηκαν οι Aberdeen City και η Stirling οι οποίες έχουν πολύ διαφορετικό πληθυσμό και δεν είναι γειτονικές περιοχές. Για κάθε μία περιοχή επιλέχθηκαν 5 DataZone areas στην τύχη, χρησιμοποιώντας την random.sample().

```
aberdeen = random.sample(DZ_CA[DZ_CA['LA_Name']=='Aberdeen City'].iloc[:,0].tolist(),5)
striling = random.sample(DZ_CA[DZ_CA['LA_Name']=='Stirling'].iloc[:,0].tolist(),5)
fig, ax = plt.subplots(figsize=(10, 5))
label=[]
for i in range(0,5):
    y=striling[i]
    dataplot = data[data['Code_Zone']==y][['Year','Price']]
    dataplot.sort_values(by=['Year'], inplace=True)
    label.append(data[data['Code_Zone']==y].iloc[0,1])
    ax.plot(dataplot.Year,dataplot.Price)
    ax.legend(label, title='title', bbox_to_anchor=(1.05, 1), loc='upper left')
    x_order=dataplot['Year'].sort_values().to_list()
    ax.set_xticklabels(x_order)
    ax.set_title("Prices per Year")
    ax.set_xlabel('Year')
    ax.set_ylabel('Price')
```

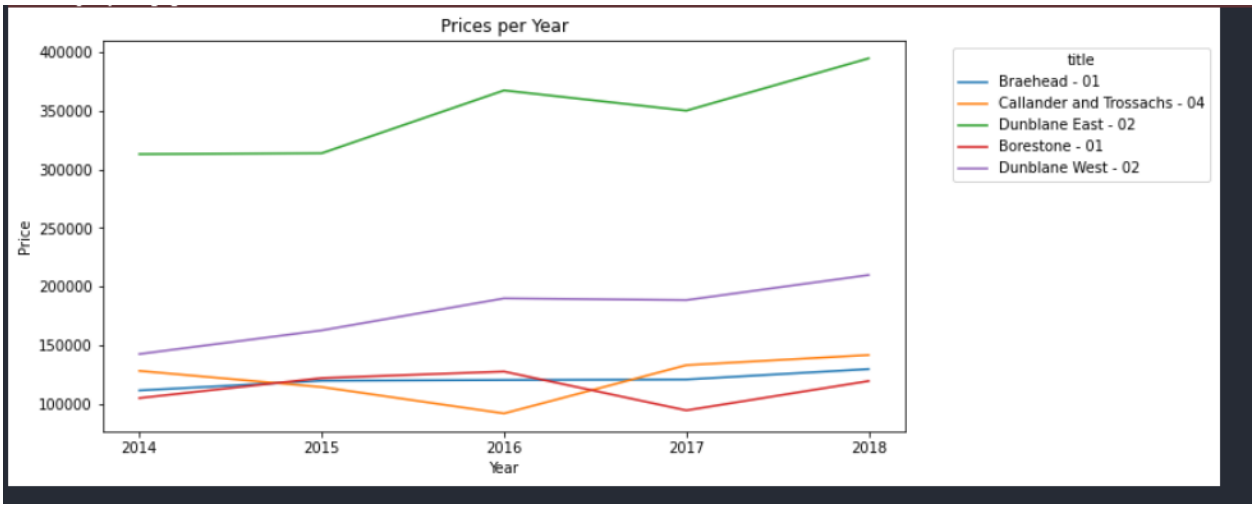
Εικόνα 17: Τρόπος δημιουργίας γραφημάτων

Aberdeen City



Εικόνα 18: Γράφημα διακύμανσης τιμών ανά έτος για περιοχές της Aberdeen City

Stirling



Εικόνα 19: Γράφημα διακύμανσης τιμών ανά έτος για περιοχές της Stirling

Στο παρακάτω διάγραμμα της περιοχής Aberdeen City διακρίνεται εύκολα ότι υπήρχε μία αυξητική τάση το 2015, την οποία όμως ακολούθησε ακόμη μεγαλύτερη μείωση το 2016. Αντίθετα στο διάγραμμα της Stirling δεν εντοπίζεται τα ίδια χαρακτηριστικά. Ως επί το πλείστων δεν παρατηρούνται έντονες μεταβολές, αλλά θα μπορούσαμε να θεωρήσουμε ότι το 2016 υπήρξε μία σχετικά αυξητική τάση. Η ίδια αντίθεση διακρίνεται και το 2018 όπου φαίνεται στο διάγραμμα της Stirling ότι μάλλον θα υπάρξει και πάλι αυξητική τάση σε αντίθεση με το διάγραμμα της Aberdeen City όπου διακρίνεται μειωτική τάση.

5.4 Διαχωρισμός δεδομένων

Για την προετοιμασία του τελικού dataframe που θα χρησιμοποιηθεί στο μοντέλο πρόβλεψης δημιουργούμε δύο επιπλέον στήλες:

- Price - 1, όπου δείχνει την τιμή των σπιτιών την προηγούμενη χρονιά
- Price - 2, όπου δείχνει την τιμή των σπιτιών δύο χρόνια πριν.

Για την δημιουργία των δύο στηλών αρχικά σορτάρουμε ανά Code_zone και ανά έτος, ώστε για κάθε row του dataframe να μπορούμε να επιλέγοντας τις προηγούμενες γραμμές να δημιουργηθούν οι νέες στήλες. Θα χρησιμοποιήσουμε τη συνάρτηση shift() η οποία μετατοπίζει το index κατά τον αριθμό περιόδου που της δίνουμε.

```
data_final=data
data_final.sort_values(by=['Code_Zone','Year'], inplace=True)
data_final['Price-1']=data_final.groupby('Code_Zone')['Price'].shift(-1)
data_final['Price-2']=data_final.groupby('Code_Zone')['Price'].shift(-2)
data_final=data_final.dropna()
```

Εικόνα 20: Χρήση της συνάρτησης shift()

Στη συνέχεια πετάμε τις περιττές στήλες, και φέρνουμε την στήλη της εξαρτημένης μεταβλητής μας (Price) στην θέση 0 του dataframe, ώστε να γίνει ο διαχωρισμός μεταξύ του X και Y μεταβλητών.

```
features=data_final.columns.drop(['Code_Zone','Name_Zone','LA_Code','LA_Name','Year'])
data_final=data_final[features]


cols_x = data_final.columns.tolist()
cols_x.remove('Price')
cols_order = ['Price']+ cols_x

data_final = data_final[cols_order]
```

Εικόνα 21: Διαχωρισμός της εξαρτημένης μεταβλητής από τις ανεξάρτητες

Το τελικό dataframe που θα χρησιμοποιηθεί για το μοντέλο πρόβλεψης είναι όπως προκύπτει παρακάτω.

```
data_final.head()
```

	Price	Births_F_ratio	Births_M_ratio	Deaths_F_ratio	Deaths_M_ratio	Educational attainment of school leavers	Ante-Natal Former Smoker	Ante-Natal Never Smoked	Ante-Natal Not Known	Fire Not Accidental	Gender PayGap full_time	Gender PayGap part_time	Female Adults with No qualifications	Male Adults with No qualifications	Price-1	Price-2
0	192917.0	6.68	5.57	0.00	4.45	6.10	3.23	80.65	0.00	0.0	13.2	-26.0	8.1	7.9	215003.0	198733.0
1	215003.0	6.66	5.55	3.33	2.22	6.23	2.70	83.78	0.00	0.0	14.8	-1.9	10.4	7.9	198733.0	149727.0
2	198733.0	5.57	6.69	3.34	5.57	6.05	5.88	82.35	0.00	0.0	9.3	-9.8	8.9	7.9	149727.0	122056.0
5	381409.0	5.97	4.78	3.58	4.78	5.48	15.62	56.25	3.12	0.0	13.2	-26.0	8.1	7.9	284539.0	161882.0
6	284539.0	2.44	4.89	6.11	7.33	5.68	15.38	69.23	3.85	0.0	14.8	-1.9	10.4	7.9	161882.0	182182.0

Εικόνα 22: Απεικόνιση του dataframe

Αφού δημιουργήθηκε το τελικό dataframe, τα δεδομένα χωρίζονται σε train data και test data. Τα train data, είναι το σύνολο των δεδομένων που θα χρησιμοποιηθούν για να εκπαιδευτεί το μοντέλο. Η στήλη y είναι η εξαρτημένη μεταβλητή, την οποία θα προσπαθήσει να προβλέψει το μοντέλο. Έτσι για κάθε πρόβλεψη (εκτιμώμενη τιμή της στήλη y) θα υπολογίζεται το σφάλμα της συνάρτησης απώλειας (loss function) και θα αναπροσαρμόζονται τα κέντρα βάρους των νευρώνων. Τα test data είναι το σύνολο δεδομένων που θα χρησιμοποιηθεί για να αξιολογήσουμε την ποιότητα και την ακρίβεια του μοντέλου. Έτσι από το συνολικό dataset των δεδομένων επιλέγουμε το 70% να χρησιμοποιηθεί ως train data, ενώ το υπόλοιπο 30% των δεδομένων θα χρησιμοποιηθεί ως test data για την αξιολόγηση.

5.5 Μοντέλο Πρόβλεψης

5.5.1 LSTM

5.5.1.1 Loss function

Για τις παραμετροποίηση του LSTM μοντέλου, αρχικά θα έπρεπε να επιλεγεί η κατάλληλη συνάρτηση απώλειας (loss function). Στα προβλήματα κατηγοριοποίηση οι πιο συνηθισμένες και ευρέως χρησιμοποιούμενες συναρτήσεις απώλειας είναι η Cross - Entropy, η Hinge loss, και η Squared hinge loss.

Η Cross entropy είναι η μία από τις πιο δημοφιλής συναρτήσεις απώλειας και μάλιστα είναι η προεπιλεγμένη συνάρτηση που χρησιμοποιείται στα περισσότερα μοντέλα. Οι πιο συνηθισμένες υποκατηγορίες είναι η binary cross entropy και η categorical cross- entropy. Η categorical cross entropy χρησιμοποιείται κυρίως σε multi class classification προσεγγίσεις. Εκεί υπάρχει n αριθμός κλάσεων και επιλέγεται η κλάση που ανάμεσα σε n πιθανότητες εμφανίζει τη μεγαλύτερη. Όταν το πρόβλημα κατηγοριοποίησης παίρνει μόνο δύο τιμές {0,1}, τότε πιο συχνά επιλέγεται η binary Cross - Entropy ως καταλληλότερη συνάρτηση απώλειας. Σε αυτήν την περίπτωση το πρόβλημα προσεγγίζεται βρίσκοντας την πιθανότητα η προβλεπόμενη τιμή να ανήκει στην μία κλάση (κατηγορία). Οπότε αντίστοιχα αποφασίζεται η τιμή 0 ή 1 σύμφωνα με τη μέγιστη πιθανότητα (maximum likelihood). Με τη χρήση λοιπόν της binary Cross- Entropy, κατά τη διάρκεια εκπαίδευσης του μοντέλου, υπολογίζεται η διαφορά μεταξύ της προβλεπόμενης και της πραγματικής πιθανότητας προκειμένου να προβλεφθεί μία συγκεκριμένη κατηγορία (κλάση). Σκοπός της συνάρτησης είναι να ελαχιστοποιηθεί το σφάλμα που προκύπτει από τη μέση διαφορά μεταξύ των πιθανοτήτων κατανομών (Yeung, Sala, Schonlieb, Rundo, 2021)

Στην αντίθετη περίπτωση υπάρχει η συνάρτηση Hinge, η οποία χρησιμοποιείται κυρίως όταν οι τιμές του προβλήματος κατηγοριοποίησης παίρνουν τιμές {-1,1}. Η hinge συνάρτηση προσπαθεί να προβλέψει το σωστό πρόσημο που θα έχει η προβλεπόμενη τιμή. Για το σκοπό αυτό όταν υπάρχουν διαφορές στο πρόσημο μεταξύ της προβλεπόμενης και της πραγματικής τιμής τότε εκχωρούνται σφάλματα στη συνάρτηση απώλειας. Η συνάρτηση Hinge

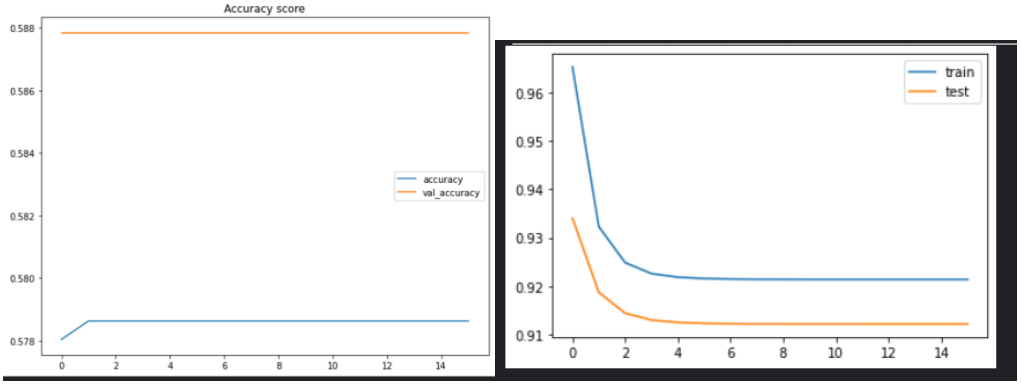
χρησιμοποιείται κατά κόρον σε μοντέλα SMV (Support vector machine) (Ozyildirim, Kiran, 2021).

Τέλος μία ακόμη συχνά χρησιμοποιούμενη συνάρτηση απώλειας είναι η squared hinge loss. Πρόκειται στην ουσία για μία παραλλαγή της hinge συνάρτηση. Η διαφορά είναι ότι υπολογίζει το τετράγωνο της απώλειας της hinge συνάρτησης. Με αυτό τον τρόπο εξομαλύνεται το αποτέλεσμα της συνάρτησης και είναι πιο εύκολο υπολογίσιμη. Συχνά η squared hinge συνάρτηση φέρνει καλύτερα αποτελέσματα από την απλή hinge συνάρτηση. Ωστόσο και σε αυτήν την περίπτωση θα πρέπει η προβλεπόμενη τιμή να παίρνει τιμές {-1,1} (Ozyildirim, Kiran, 2021).

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν και τις τέσσερις διαφορετικές συναρτήσεις απώλειας και έγινε η σύγκριση των αποτελεσμάτων για να επιλεχθεί η συνάρτηση με τα βέλτιστα αποτελέσματα. Στο συγκεκριμένο μοντέλο, καθώς αρχικά δεν είναι ένα μοντέλο SVM, και έπειτα οι τιμές που μπορεί να πάρει η κλάση είναι μεταξύ {0,1}, αναμένεται να έχει καλύτερη απόδοση η binary Cross - Entropy. Έτσι αφού έγινε η εκπαίδευση του μοντέλου με κάθε μία από τις παραπάνω συναρτήσεις, καλούμε με το history callback τις μετρήσεις accuray και loss που έχουν προέλθει από τα train και τα test data αντίστοιχα.

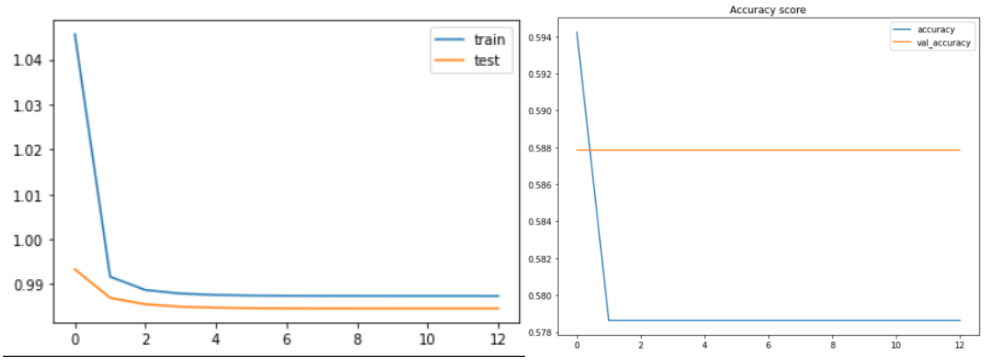
Όπως βλέπουμε και στα παρακάτω διαγράμματα, οι συνάρτηση hinge και squared hinge δεν αποδίδουν καθόλου αποτελεσματικά για το συγκεκριμένο μοντέλο. Η διαφορά μεταξύ των συναρτήσεων categorical classification και binary classification δεν είναι πολύ μεγάλη, ωστόσο , όντως η συνάρτηση binary classification φαίνεται να αποδίδει το βέλτιστο.

Συνάρτηση απώλειας: Hinge



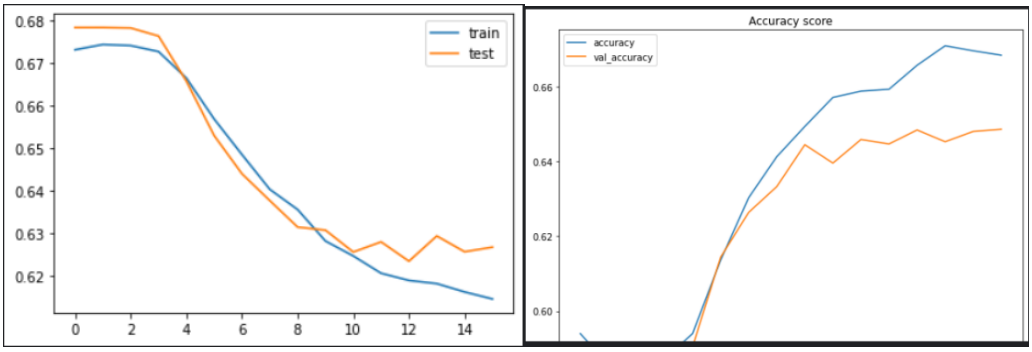
Εικόνα 23: Γράφημα της συνάρτηση απώλειας (Hinge) σε train και test data

Συνάρτηση απώλειας: Square Hinge



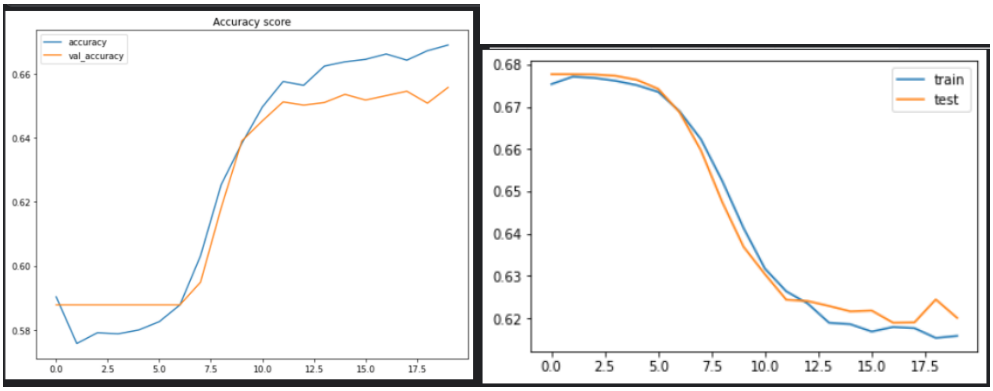
Εικόνα 24: Γράφημα της συνάρτηση απώλειας (Square Hinge) σε train και test data

Συνάρτηση απώλειας: Categorical crossentropy



Εικόνα 25: Γράφημα της συνάρτηση απώλειας (Categorical crossentropy) σε train και test data

Συνάρτηση απώλειας: Binary categorical crossentropy



Εικόνα 26: Γράφημα της συνάρτηση απώλειας (Binary categorical crossentropy) σε train και test data

5.5.1.2 Activation Function

Η συνάρτηση ενεργοποίησης ενός μοντέλου νευρωνικού δικτύου είναι μία εξίσου σημαντική απόφαση, καθώς από τη συνάρτηση που θα επιλεχθεί στο τελευταίο επίπεδο (Output Layer) θα καθοριστεί και ο τύπος των προβλέψεων του μοντέλου. Σε κάθε κόμβο εισέρχεται ένα διάνυσμα τιμών, το οποίο πολλαπλασιάζεται με τα βαρύ του κόμβου. Στη συνέχεια οι νέες αυτές τιμές αθροίζονται μεταξύ τους. Το σταθμισμένο αυτό άθροισμα που θα προκύψει ως αποτέλεσμα (weighted sums), θα μετατραπεί με την συνάρτηση ενεργοποίησης σε παραγόμενο αποτέλεσμα (output). Η συνάρτηση ενεργοποίησης στην ουσία καθορίζει τον τρόπο με το οποίο το διάνυσμα με τα βάρη που εισέρχεται σε ένα κόμβο θα μετατραπεί σε αποτέλεσμα.

Κατ' επέκταση η επιλογή της συνάρτησης ενεργοποίησης επηρεάζει άμεσα την απόδοση του μοντέλου. Ένα μοντέλο μπορεί να έχει μεγάλη ακρίβεια στις προβλέψεις του και με την αλλαγή της συνάρτησης ενεργοποίησης να βελτιωθεί σημαντικά το αποτέλεσμα.

Στα νευρωνικά δίκτυα τόσο τα κρυμμένα επίπεδα (hidden layers) αλλά και το τελικό επίπεδο που κάνει τις προβλέψεις (output layer) χρησιμοποιούν μία συνάρτηση ενεργοποίησης. Συνήθως όλα τα κρυμμένα επίπεδα χρησιμοποιούν την ίδια συνάρτηση ενεργοποίησης η οποία όμως διαφέρει από τη συνάρτηση του τελικού επιπέδου.

Ως κρυμμένο επίπεδο είναι όλα τα επίπεδα που δεν παίρνουν άμεσα το input του μοντέλου, αλλά το λαμβάνουν μέσω ενός άλλου επιπέδου, και επίσης δεν παράγουν απευθείας το output του μοντέλου, αλλά μεταβιβάζουν το output τους σε άλλο επίπεδο. Στις περιπτώσεις λοιπόν των κρυφών μοντέλων συχνά η καλύτερη επιλογή είναι μία μη γραμμική συνάρτηση, ώστε το μοντέλο να είναι σε θέση να εκπαιδευτεί με πιο σύνθετες συναρτήσεις.

Rectified Linear Unit (Relu)

Η πιο συνηθισμένη συνάρτηση ενεργοποίησης είναι η ReLu και εκφράζεται από την

$\sigma(x) = \max(x, 0)$

Η συνάρτηση επιστρέφει τον μεγαλύτερο αριθμό ανάμεσα στο x και το μηδέν. Στη περίπτωση δηλαδή που το x είναι αρνητικός αριθμός τότε το αποτέλεσμα της σ(x) θα είναι μηδέν, ενώ αν το x είναι θετικός αριθμός τότε θα επιστραφεί το x ως αποτέλεσμα. Δεδομένου του τρόπου υπολογισμού της είναι προτιμότερο τα δεδομένα να μετατρέπονται σε κλίμακα (0,1) (normalization) (Parisi, Neagu, Ma, Campean, 2021).

Sigmoid

Η σιγμοειδής συνάρτηση πήρα το όνομα της από τον τρόπο του γραφήματος της, το οποίο μοιάζει με το γράμμα σίγμα (S - Shape). Διαφέρει στον τρόπο υπολογισμού σε σχέση με τη ReLu. Αντί να αποδώσει ως αποτέλεσμα τον μεγαλύτερο αριθμό, η σιγμοειδής δίνει πάντα ένα αριθμό της κλίμακας (0,1), ο οποίος εκφράζει μία πιθανότητα. Με αυτόν τον τρόπο όσο πιο πιθανών είναι ένα γεγονός τόσο η τιμή που θα προκύψει από τη σιγμοειδή συνάρτηση θα είναι πιο κοντά στη μονάδα (1). Ενώ στην αντίθετη περίπτωση όσο μικρότερη είναι η πιθανότητα τόσο και η τιμή της συνάρτησης θα πλησιάζει στο μηδέν (0).
Η μαθηματική απεικόνιση της συνάρτησης εκφράζει παρακάτω:

$$1.0 / (1.0 + e^{-x})$$

Ωστόσο το μοντέλο είναι πιο επιρρεπείς σε πρόβλημα όπως το vanishing gradients.

Hyperbolic tangent (tanh)

Η συνάρτηση tanh έχει αρκετά κοινά στοιχεία με τη συνάρτηση sigmoid, με πρώτο μάλιστα ότι έχει το ίδιο σχήμα ραφήματος (S - shape). Η συγκεκριμένη συνάρτηση επιδέχεται ως input οποιαδήποτε τιμή, και επιστρέφει ως αποτέλεσμα μία τιμή της κλίμακας (-1,1). Όσο μεγαλύτερη (άρα θετικό νούμερο) είναι η τιμή εισόδου που δέχεται τόσο η επιστρεφόμενη τιμή θα πλησιάζει τη μονάδα(1). Στην αντίθετη περίπτωση όσο μικρότερη είναι η τιμή εισόδου (άρα αρνητική τιμή) τόσο το αποτέλεσμα που θα επιστραφεί θα είναι κοντά στο -1.
Η μαθηματική απεικόνιση της συνάρτησης εκφράζει παρακάτω:

$$(e^{x} – e^{-x}) / (e^{x} + e^{-x})$$

Όπως αναφέρθηκε και παραπάνω ενώ χρησιμοποιείται μία κοινή συνάρτηση ενεργοποίησης για όλα τα κρυμμένα επίπεδα ενός μοντέλου, αυτή διαφοροποιείται από τη συνάρτηση ενεργοποίησης που χρησιμοποιείται στο τελικό επίπεδο που παράγεται το αποτέλεσμα (output layer). Οι πιο σύνηθες επιλεγόμενες συναρτήσεις είναι οι: Lineal, Logistic και Softmax.

Lineal

Η συγκεκριμένη συνάρτηση δεν μεταβάλλει καθόλου το σταθμισμένο άθροισμα (weighted sums) που επιδέχεται ως τιμή εισόδου. Είναι πολύ εύκολη στη χρήση του μοντέλου ωστόσο δυσκολεύεται να χειριστεί πιο περίπλοκες συναρτήσεις ώστε να φέρει μία ικανοποιητική απόδοση. Για το λόγο αυτή η γραμμική συνάρτηση προτιμάται κυρίως σε προβλήματα regression και μάλιστα στο τελικό επίπεδο.

Softmax

Η συνάρτηση softmax επιστρέφει ένα διάνυσμα του οποίου οι τιμές αθροίζουν στη μονάδα, όπως οι πιθανότητες.

$$e^x / \text{sum}(e^x)$$

Ενώ σε προβλήματα regression η Lineal φαίνεται καλύτερη επιλογή, στα προβλήματα κατηγοριοποίησης αντιδρούν καλύτερα οι sigmoid και η softmax. Όταν υπάρχουν δύο μόνο κλάσεις τότε η πιο σύνηθες επιλογή είναι η sigmoid, ενώ όταν υπάρχουν περισσότερες από δύο κλάσεις τότε καλύτερο είναι να χρησιμοποιηθεί η softmax.

Στο μοντέλο LSTM συνήθως χρησιμοποιείται η σιγμοειδής συνάρτηση για τις συνδέσεις μεταξύ των κρυμμένων επιπέδων και την συνάρτηση tanh για την επίπεδο της εξόδου. Η ReLu βέβαια προτιμάται αρκετά σε πολυστρωματικά μοντέλα.

Εφαρμογή

Δεδομένου όλων των παραπάνω το συγκεκριμένο μοντέλο LSTM καθώς έχει μόνο δύο κλάσεις, επιλέχθηκε να χρησιμοποιηθεί η συνάρτηση softmax και sigmoid. Χρειάστηκε

ωστόσο να γίνει και Trial and error όπου και διαπιστώθηκε ότι το μοντέλο βελτίωνε τις προβλέψεις του με το συγκεκριμένο συνδυασμό συναρτήσεων. Στα κρυφά επίπεδα η εντολή της βιβλιοθήκης keras της Python είναι η παράμετρος activation='softmax', ενώ για το τελευταίο επίπεδο έχουμε την παράμετρο activation='sigmoid'.

5.5.1.3 Optimizer

Stochastic Gradient Descent (SGD)

Ο συγκεκριμένος βελτιστοποιητής προσπαθεί μέσω της επανάληψης να εντοπίσει το μέγιστο είτε το ελάχιστον σφάλμα. Η μέθοδος SGD δεν εξαρτάται από το μέγεθος των ρυθμών εκμάθησης (Jentzen, Wurstemberger, 2020). Για την επιλογή του συγκεκριμένου Optimizer είναι σημαντικό να ελεγχθεί η αντικειμενική συνάρτηση του μοντέλου. Αν για το μοντέλο έχει επιλεχθεί μία συνάρτηση η οποία δεν είναι κυρτή, τότε ο SGD δεν αποτελεί τη βέλτιστη επιλογή. Στην περίπτωση της μη κυρτή συνάρτηση ο SGD θα συγκλίνει προς το τοπικό ελάχιστο (ή μέγιστο αντίστοιχο) χωρίς να εντοπίσει το ολικό ελάχιστο (ή μέγιστο), με αποτέλεσμα να μην είναι αποδοτικό.

Adam

Ο βελτιστοποιητής Adam περιγράφει πως γίνεται η προσαρμογή των βαρών μετά από κάθε επανάληψη με τέτοιο τρόπο ώστε να παραμείνουν ελεγχόμενα και αμερόληπτα. Με την χρήση του Adam υπολογίζονται οι προσαρμοστικοί ρυθμοί εκμάθησης για διαφορετικές παραμέτρους. Το όνομα του Adam το πήρε από την προσαρμοστικότητα των ρυθμών εκμάθησης (Adaptive learning rates) (Kingma, Ba, 2015). Η συγκεκριμένη μέθοδος είναι από τις πιο δημοφιλείς και έχει αρκετά πλεονεκτήματα. Το σημαντικότερο ίσως είναι ότι έχει υψηλή υπολογιστική απόδοση ενώ ταυτόχρονα δεν απαιτεί μεγάλη μνήμη, και για αυτό το λόγο είναι η πλέον κατάλληλη σε προβλήματα μεγάλων δεδομένών ή πολλών παραμέτρων (Chang, Zhang, Chen, 2019). Η μέθοδος Adam χρησιμοποιεί δύο διαφορετικές τεχνικές gradient descent, την momentum, και την RMSP (Root Mean Square Propagation). Η μέθοδος Momentum χρησιμοποιείται εκτενώς για την βελτίωση των αλγορίθμους backpropagation όπως δηλαδή ο LSTM, και χρησιμοποιεί τους εκθετικούς σταθμικούς

μέσους. Από την άλλη πλευρά η μέθοδος RMSP αποτελεί μία βελτίωση του AdaGrad μοντέλου και βασίζεται στον εκθετικό κινητό μέσο όρο. Έτσι με την χρήση της μεθόδου Adam το μοντέλο παίρνει τα πλεονεκτήματα και από τις δύο τεχνικές (Salem, Kabeel, El-Said, Elzeki, 2021)

Εφαρμογή

Με βάση λοιπόν την παραπάνω ανάλυση στο μοντέλο LSTM έχει χρησιμοποιηθεί ο Adam ως optimizer. Είναι καταλληλότερος για τα προβλήματα της οπίσθιας ανατροφοδότησης, όπως το μοντέλο LSTM, και ταυτόχρονα μπορεί να χειρίζεται εξαιρετικά τα μεγάλα δεδομένα. Στην βιβλιοθήκη keras της Python αρκεί να θέσουμε την παραμετρό optimizer = 'adam' στην εντολή compile του μοντέλου, από όπου προκύπτει το τελικό output.

5.5.2 XGB

Τα μοντέλα xgboost λειτουργούν καλύτερα όταν οι εξαρτημένες μεταβλητές (y) είναι αριθμητικές. Εάν πρόκειται για regression πρόβλημα, οι μεταβλητές y είναι by default συνεχείς αριθμητικές μεταβλητές. Στην περίπτωση όμως του classification, συνίσταται η κάθε τιμή (κλάση) να αντιστοιχίσετε σε numeric values. Για παράδειγμα το μοντέλο προσπαθούσε να προβλέψει το φύλλο των γεννήσεων τότε θα λάμβανε δύο τιμές "Άντρας" και "Γυναίκα". Σε μία τέτοια περίπτωση η μία μεταβλητή θα κατηγοριοποιούνται ως μηδέν και ένα. Στο συγκεκριμένο case που ελέγχει η παρούσα εργασία έχει γίνει ήδη η μετατροπή της εξαρτημένης μεταβλητής σε (0,1), οπότε δεν χρειάζεται περαιτέρω επεξεργασία.

Για την εκπαίδευση αλλά και την αξιολόγηση του xgb μοντέλου, θα χρησιμοποιηθεί το ίδιο σύνολο των δεδομένων που χρησιμοποιήθηκε στο lstm μοντέλο. Οπότε και σε αυτό το case θα έχουμε το 70% των data να χρησιμοποιείται ως train dataset και το υπόλοιπο 30% ως test dataset.

5.5.2.1 Tuning

Booster

Με την παράμετρο booster καθορίζεται ουσιαστικά το είδους της εκπαίδευσης που θα λάβει το μοντέλο και είναι αυτό που θα τρέξει σε κάθε επανάληψη ο αλγόριθμος. Οι πιο συνηθισμένες μέθοδοι booster είναι οι gbtree και η dart για την περίπτωση των δέντρων, και ο gblinear για την περίπτωση του γραμμικού booster.

Στην μέθοδος dart προκειμένου να αποφευχθεί το overfitting, αποκόβονται δέντρα. Για το λόγο αυτό, όταν χρησιμοποιείται η συγκεκριμένος μέθοδος συνίσταται να ορίζεται επίσης και ένα σημείο στο οποίο να σταματήσει το drop trees. Αυτό ορίζεται με την παράμετρο rate_drop, όπου η default τιμή της είναι 0.0.
Η gbtree είναι η προεπιλεγμένη μέθοδος boostoer, και αυτή που χρησιμοποιούμε και εμείς στο παρόν μοντέλο.

Objective

Εφόσον πρόκειται για classification, αντίστοιχα όπως και στο LSTM μοντέλο έτσι και στο xgb, τα δύο είδη συναρτήσεων είναι το Binary Classification (binary:logistic) και το Multi-Class Classification (multi:softprob).Σε περίπτωση που δεν ορίσουμε ποια συνάρτηση θέλουμε, τότε θα επιλεχθεί μία συνάρτηση από το ίδιο το μοντέλο σύμφωνα με τα label data που θα λάβει. Στο παρόν μοντέλο, καθώς όπως αναφέραμε έχει μόνο δύο κλάσεις, ορίστηκε η binary objective function.

Learning rate *(eta)*

Η παράμετρος eta εκφράζει το βαθμό εκμάθησης του μοντέλου. Το xgb δημιουργεί συνεχώς νέα δέντρα ώστε να ελαχιστοποιήσει το σφάλμα των καταλοίπων. Οι πιο συνηθισμένες τιμές είναι 0.1, 0.01, 0.001. Στον συγκεκριμένο μοντέλο χρησιμοποιήθηκε το 0.01.

Max depth

Για την εύρεση της βέλτιστης τιμής του μέγιστου βάθους που μπορεί να καλλιεργηθεί ένα δέντρο, χρησιμοποιήθηκε η GridSearchCV. Με αυτή τη μέθοδο, μπορούμε να θέσουμε διαφορετικές τιμές μιας παραμέτρου και να βρεθεί η βέλτιστη τιμή, στην οποία αποδίδει καλύτερα το μοντέλο. Για να γίνει αυτό η GridSearchCV χρησιμοποιεί διαφορετικές τιμές

της παραμέτρου και υπολογίζει την απόδοση κάθε μίας από αυτές. Σε περίπτωση που χρησιμοποιηθούν περισσότερες από μία μεταβλητές, υπολογίζει την απόδοση για κάθε έναν διαφορετικό συνδυασμό των τιμών των μεταβλητών. Στην συνέχεια βρίσκει ποια τιμή (ή ποιος συνδυασμός τιμών) είχε την καλύτερη απόδοση και ορίζει την βέλτιστη τιμή της μεταβλητής. Στην συγκεκριμένη περίπτωση τέθηκε το εύρος του βάθους των δέντρων ως 3, 5 και 7. Όπως βλέπουμε και παρακάτω προέκυψε ότι η βέλτιστη τιμή του max_depth είναι το 3.

```
model_x = XGBClassifier(objective='binary:logistic', booster='gbtree', learning_rate=0.01, eval_metric='mlogloss', use_label_encoder=False)
max_depth = range(3,5, 7)
print(max_depth)
param_grid = dict(max_depth=max_depth)
#kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=7)
grid_search = GridSearchCV(model_x, param_grid, scoring="accuracy", cv=10, verbose=1)
grid_result = grid_search.fit(train_X_xgb, train_y_xgb)
# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
```

```
range(3, 5, 7)
Fitting 10 folds for each of 1 candidates, totalling 10 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  10 out of  10 | elapsed:    8.5s finished
Best: 0.487403 using {'max_depth': 3}
```

Εικόνα 27: Εύρεση βέλτιστων παραμέτρων για το xgboost μοντέλο

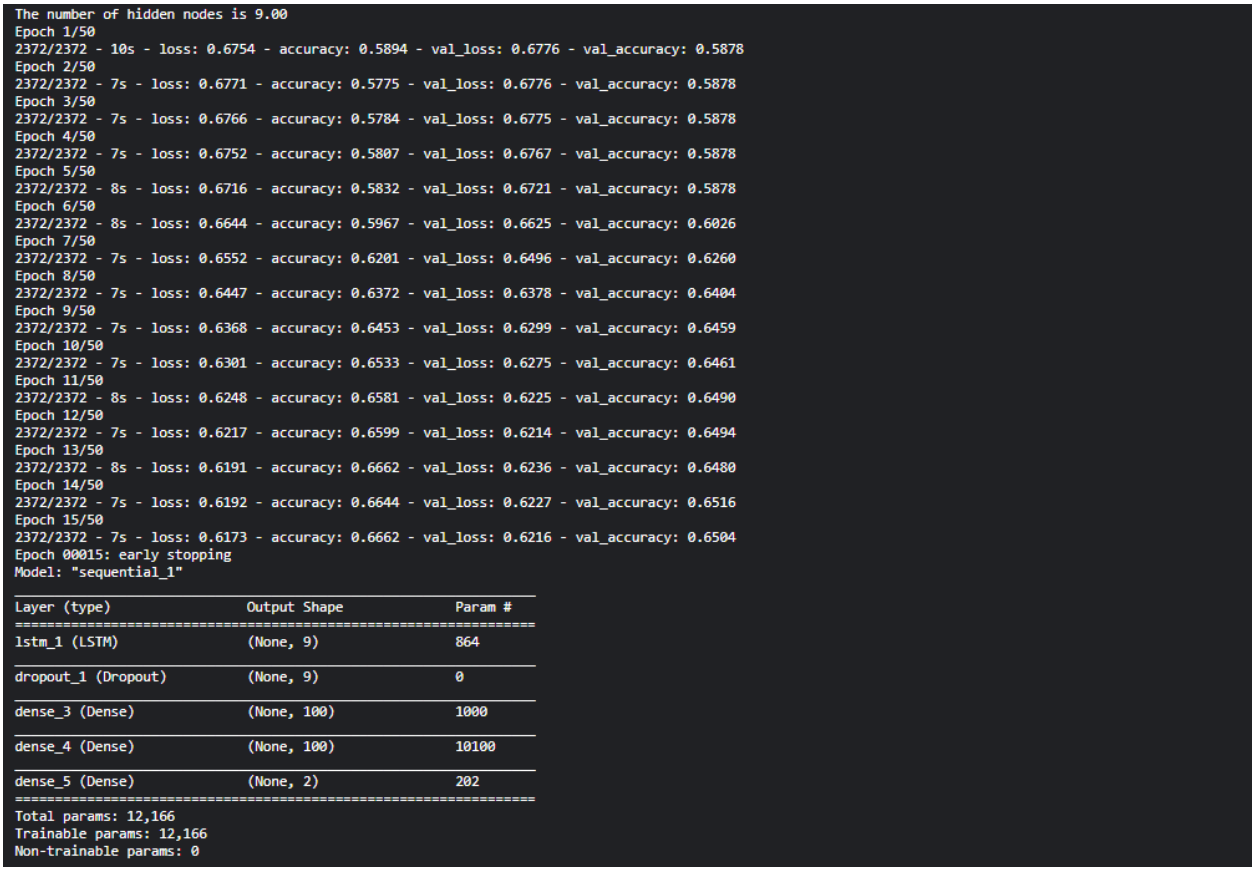
5.5.3 Εφαρμογή

Σύμφωνα με όλες τις παραπάνω παραμέτρους, όπως αυτές εξηγήθηκαν, προέκυψαν τα δύο μοντέλα.

5.5.3.1 LSTM Model

```
n_outputs=train_y.shape[1]

nodes = int(2/3*(train_X_lstm.shape[1]*train_X_lstm.shape[2]))
print('The number of hidden nodes is %.2f' %  (nodes))

model_lstm = Sequential()
model_lstm.add(LSTM(nodes, input_shape=(train_X_lstm.shape[1], train_X_lstm.shape[2])))  #input_shape=(n_features,look_back)
model_lstm.add(Dropout(0.2))
model_lstm.add(Dense(100, activation='softmax'))
model_lstm.add(Dense(100, activation='softmax'))
model_lstm.add(Dense(n_outputs, activation='sigmoid'))
model_lstm.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

es = EarlyStopping(monitor='val_loss', mode='min', verbose=2, patience=3)
history = model_lstm.fit(train_X_lstm, train_y, batch_size=5, validation_data=(test_X_lstm, test_y), epochs=50,callbacks=[es], verbose=2,  shuffle=False)

model_lstm.summary()
```

Εικόνα 28: Δημιουργία LSTM μοντέλου



Εικόνα 29: Εκπαίδευση LSTM μοντέλου

Κατά την εκπαίδευση του μοντέλου βλέπουμε αρχικά ότι ενώ τα έκανε 50 epoch, σταμάτησε στην 15, καθώς έχουμε ορίσει το early stopping με patience=3. Αυτό πρακτικά σημαίνει ότι όταν για τρεις συνεχόμενες επαναλήψεις δεν παρατηρηθεί βελτίωση στο μοντέλο, τότε θα σταματήσει, θεωρώντας ότι έχει ήδη βρει τη βέλτιστη τιμή του. Πράγματι, βλέπουμε ότι ενώ μέχρι την 12 επανάληψη παρατηρείται μία βελτίωση στις αποδόσεις του μοντέλου, μετά την 12 σταδιακά εξασθενεί.

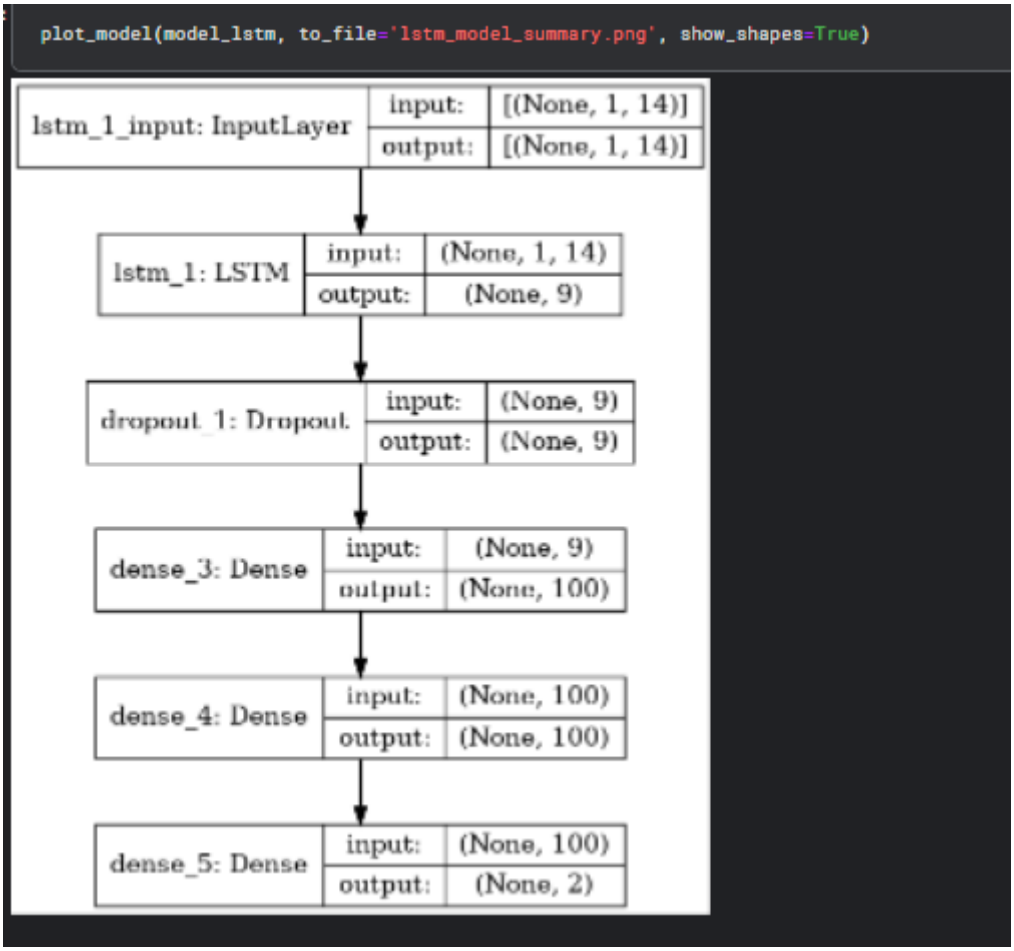
Στο summary του lstm μοντέλου, κάθε γραμμή αντιπροσωπεύει και ένα επίπεδο (layer) του lstm μοντέλου. Έτσι μπορούμε να μία ματιά να δούμε όλα τα επίπεδα που δημιουργήσαμε παραπάνω καθώς και το output shape που παράγεται από κάθε layer. Στην στήλη “Output Shape” του κάθε επιπέδου μπορούμε να δούμε το output shape που παράγει κάθε layer και το οποίο χρησιμοποιείται ως input για το επόμενο επίπεδο, πέραν φυσικά του τελευταίου

layer, του οποίου το output είναι η τελική πρόβλεψη. Για τα Dense layers ο τρόπος που γίνεται ο υπολογισμός των παραμέτρων είναι ο ακόλουθος:

$$\text{Param} = \text{output\_number} * (\text{input\_number} + 1)$$

Στην στήλη Param # παρουσιάζεται αντίστοιχα ο αριθμός των παραμέτρων που χρησιμοποιεί κάθε layer για την εκπαίδευση. Τέλος βλέπουμε ότι στο παρόν lstm μοντέλο έχουν συνολικά χρησιμοποιηθεί 12.166 παράμετροι κατά την εκπαίδευση του, και ότι δεν υπάρχουν παράμετροι που δεν χρησιμοποιήθηκαν (Non - trainable param = 0).

Μπορούμε ακόμη να δούμε αναλυτικά την αρχιτεκτονική του lstm μοντέλου και την σύνδεση των layers από το ακόλουθο σχεδιάγραμμα.



Εικόνα 30: Αρχιτεκτονική LSTM μοντέλου

5.5.3.2 XGB Classifier

Ακολουθείς δημιουργούμε το xgb model, όπου στο παραγόμενο output βλέπουμε τις παραμέτρους που χρησιμοποιήθηκαν κατά την διάρκεια της εκπαίδευσης του μοντέλου. Κάποιες παράμετροι παίρνουν την τιμή που έχουμε θέσει, όπως για παράδειγμα το max_depth = 3, ενώ όσες δεν έχουν ορισθεί από εμάς, παίρνουν την default τιμή τους, όπως για παράδειγμα η

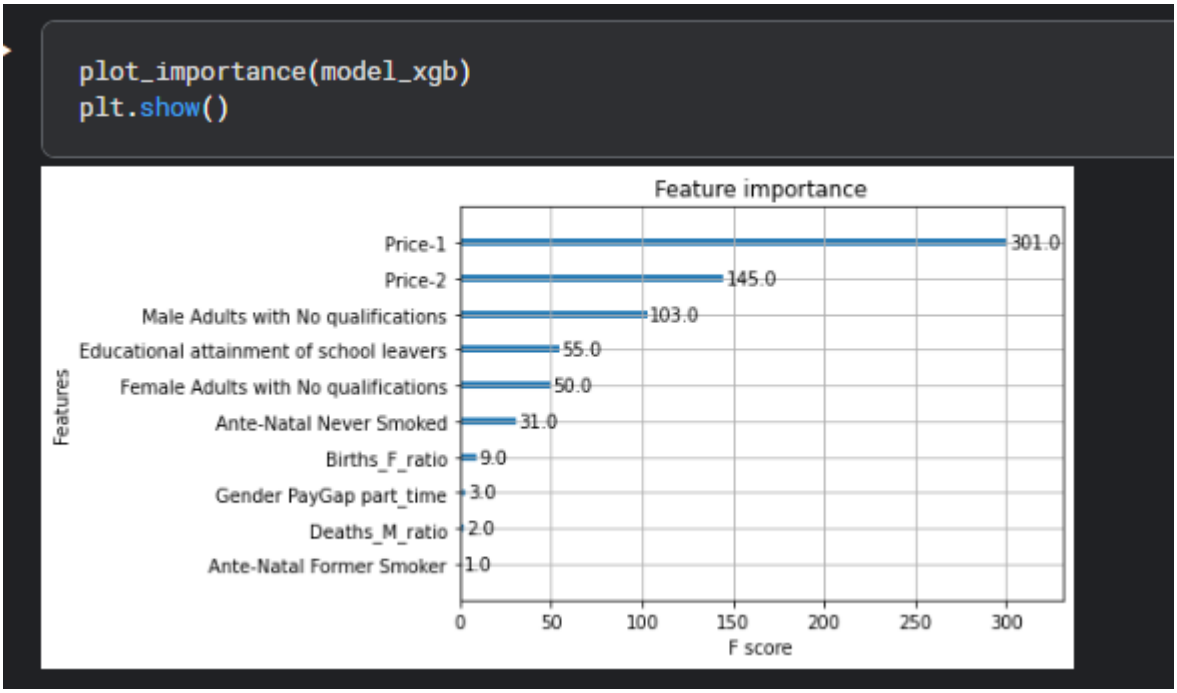
Colsample_by_tree που δείχνει την αναλογία των στηλών που έχουν επιλεγεί (με τυχαία επιλογή) και έχουν χρησιμοποιηθεί για την εκπαίδευση του κάθε δέντρου. Η δειγματοληψία πραγματοποιείται για κάθε δέντρο που σχηματίζεται μία φορά. Όλες οι μεταβλητές colsampe_by έχουν την default τιμή τους ίση με τη μονάδα

```
#XGB model

train_X_xgb, train_y_xgb = train[:, 1:], train[:,0]
test_X_xgb, test_y_xgb = test[:, 1:], test[:, 0]

model_xgb = XGBClassifier(objective='binary:logistic', booster='gbtree', learning_rate=0.01, max_depth=3, eval_metric='mlogloss',  use_label_encoder=False)
model_xgb.fit(train_X_xgb, train_y_xgb)
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              eval_metric='mlogloss', gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.01, max_delta_step=0,
              max_depth=3, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=4,
              num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', use_label_encoder=False,
              validate_parameters=1, verbosity=None)
```

Εικόνα 31: Δημιουργία xgboost μοντέλου

Ένα επιπλέον πλεονέκτημα του μοντέλου xgboost είναι ότι μπορεί να υπολογίσει τη σημαντικότητα της κάθε μεταβλητής κατά των σχεδιασμό των δέντρων. Ένα χαρακτηριστικό αποκτά μεγαλύτερη σημασία κατά τη διαδικασία εκπαίδευσης των δέντρων όσο περισσότερο χρησιμοποιείται. Καθώς κάθε χαρακτηριστικό έχει έναν συγκεκριμένο δείκτη F score, μπορεί εύκολα να γίνει η σύγκριση μεταξύ τους και να παραχθεί το παρακάτω γράφημα της σημαντικότητας (Feature Importance).

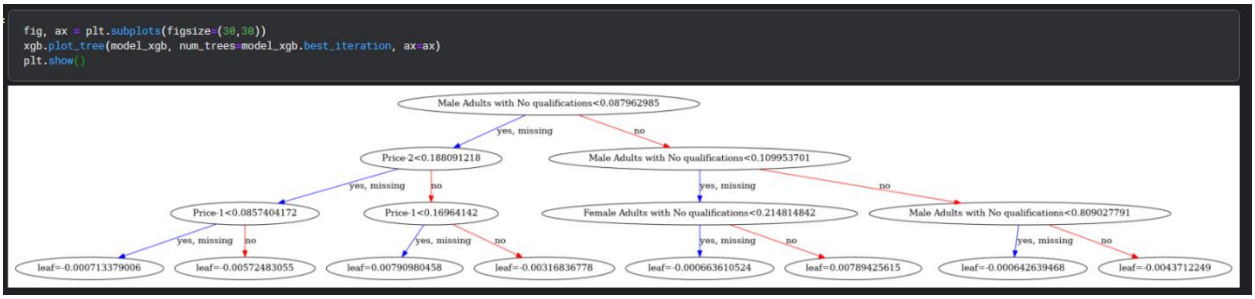


Εικόνα 32: Γράφημα απεικόνισης σημαντικότητας μεταβλητών του xgboost μοντέλου

Βλέπουμε ότι οι δύο πιο σημαντικές μεταβλητές είναι οι τιμές των προηγούμενων μεταβλητών, το οποίο είναι αναμενόμενο στις χρονοσειρές δεδομένων. Είναι γεγονός ότι όταν πρόκειται για την μεταβολή των τιμών, η τιμή της προηγούμενης χρονικής περιόδου επηρεάζει άμεση την τιμή της επόμενης χρονικής περιόδου. Άλλωστε και η κλάση (y) (αύξηση ή μείωση της τιμής) την οποία προσπαθούμε να προβλέψουμε έχει σχηματιστεί χρησιμοποιώντας ώς παράμετρο την τιμή της προγενέστερης περιόδου. Μάλιστα η τιμή της προηγούμενης χρονιάς (Price - 1) φαίνεται ότι έχει αρκετά μεγαλύτερη σημασία από την τιμή των δύο προηγούμενων ετών (Price - 2).

Επίσης εξίσου διακριτό είναι το γεγονός ότι οι τιμές επηρεάζονται αρκετά από τα εκπαιδευτικά χαρακτηριστικά του πληθυσμού των περιοχών. Για το λόγο αυτό βλέπουμε ότι τα αμέσως τρία επόμενα χαρακτηριστικά που φαίνεται να είναι σημαντικά διακρίνουν το εκπαιδευτικό επίπεδο των κατοίκων των περιοχών. Μάλιστα ενδιαφέρον αποτελεί το γεγονός ότι το ποσοστό των ανδρών με χαμηλή ή καθόλου προσόντα έχει σχεδόν διπλάσιο δείκτη σημαντικότητας από το αντίστοιχο ποσοστό των γυναικών με χαμηλά ή καθόλου προσόντα.

Τέλος μπορούμε και εδώ να δούμε την αρχιτεκτονική του xgb μοντέλου και πως έχει σχεδιαστεί το τελικό δέντρο, σύμφωνα με τις πιο σημαντικές μεταβλητές.



Εικόνα 33: Αρχιτεκτονική xgboost μοντέλου

5.6 Αξιολόγηση Μοντέλων

Για την αξιολόγηση των classification μοντέλων χρησιμοποιούνται οι παρακάτω δείκτες είτε με την άμεση πληροφορία που δίνει ο κάθε ένας, είτε έμμεσα, παράγοντας δηλαδή άλλες μετρήσεις συνδυαστικά.

TP (True Positive): όταν μία κλάση προβλέφθηκε ως θετική και ήταν όντως θετική. Στο συγκεκριμένο μοντέλο μεταφράζει για παράδειγμα ότι όταν ήταν μηδενική κλάση προβλέφθηκε θετικώς ως μηδενική.

TN (True Negative): όταν μία κλάση προβλέφθηκε ως αρνητική και ήταν όντως αρνητική. Στο συγκεκριμένο μοντέλο μεταφράζει για παράδειγμα ότι όταν ήταν κλάση ίση με τη μονάδα προβλέφθηκε θετικώς ως μονάδα.

FP (False Positive): όταν μία κλάση ήταν αρνητική αλλά προβλέφθηκε ως θετική. Στο συγκεκριμένο μοντέλο μεταφράζει για παράδειγμα ότι όταν ήταν μηδενική κλάση προβλέφθηκε αρνητικώς ως μονάδα.

FN (False Negative): όταν μία κλάση ήταν θετική αλλά προβλέφθηκε ως αρνητική. Στο συγκεκριμένο μοντέλο μεταφράζει για παράδειγμα ότι όταν ήταν κλάση ίση με τη μονάδα προβλέφθηκε αρνητικώς ως μηδενική.

5.6.1 Accuracy

Το πιο συνηθισμένο μέγεθος αξιολογησης ενός classification μοντέλου είναι η ακρίβεια (accuracy). Ως accuracy ορίζεται το ποσοστό των κλάσεων που έχουν προβλεφθεί επιτυχώς. Ο υπολογισμός της accuracy του μοντέλου γίνεται ως:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Έτσι για να υπολογιστούν τα evaluation metrics χρειάζεται πρώτα να γίνει η πρόβλεψη του μοντέλου (y_ predicted).

```
#****  LSTM Model  *******
#Prediction in LSTM model
trainPredict = model_lstm.predict(train_X_lstm)
testPredict = model_lstm.predict(test_X_lstm)
#Accuracy in LSTM Model
accuracy_lstm = model_lstm.evaluate(test_X_lstm, test_y, verbose=0)[1]

#****  GXB Model  *******
#Prediction in LSTM model
y_pred = model_xgb.predict(test_X_xgb)
predictions = [round(value) for value in y_pred]
#Accuracy in LSTM Model
accuracy_xgb = accuracy_score(test_y_xgb, predictions)
```

```
print('\nAccuracy LSTM model: %.2f%%' % (accuracy_lstm*100.0))

print("\nAccuracy XGB model: %.2f%%" % (accuracy_xgb * 100.0))
```

```
Accuracy LSTM model: 64.96%

Accuracy XGB model: 63.68%
```

Εικόνα 34: Σύγκριση του δείκτης accuracy μεταξύ των μοντέλων

Στην παραπάνω εικόνα φαίνεται ότι το LSTM μοντέλο έχει ελαφρώς καλύτερη ακρίβεια πρόβλεψης (64,96%) έναντι του αντίστοιχου Xgboost μοντέλου (63,68%).

Ωστόσο με γνώμονα μόνο την accuracy δεν μπορούμε να βασιστούμε ώστε να επιλεγεί το καλύτερο μοντέλο. Συχνά χρειάζεται να ελεγχθούν και άλλες αντίστοιχες τιμές μέτρησης αξιολόγησης.

5.6.2 Classification Report

Το classification report αποτελεί ένα καλό report για να αξιολογηθεί ένα classification μοντέλο, όπως αυτό που ελέγχθηκε στα πλαίσια της παρούσας εργασίας. Το συγκεκριμένο report υπολογίζει την ακρίβεια του μοντέλου (precision), το recall, και το F1 score.

Precision

Η έννοια του precision περιγραφεί ο ποσοστό των προβλέψεων που έγιναν σωστά. Μπορεί δηλαδή να απαντήσει στο ερώτημα “σε πόσους συναίβει, αυτό που προέβλεψε το μοντέλο ότι θα συμβεί”. Ο υπολογισμός του γίνεται διαιρώντας τα True Positive με το άθροισμα των True Position και False Positive:

$$\frac{TP}{sum(TP + FP)}$$

Recall

Αντίθετα η έννοια του recall εκφράζει το ποσοστό των θετικών προβλέψεων που μπόρεσε να ερμηνευθεί σωστά από το μοντέλο. Το recall απαντά στην ερώτηση “πόσους κατάφερε το μοντέλο να προβλέψει από αυτούς που όντως του συνέβη”. Ο υπολογισμός του γίνεται διαιρώντας τα True Positive με το άθροισμα των True Position και False Negative:

$$\frac{TP}{sum(TP + FN)}$$

F1 Score

Το F1 score είναι μία μετρική που συνδυάζει τις δύο προηγούμενες (Precision, recall). Συχνά όταν συγκρίνουμε τις αποδόσεις δύο οι περισσότερων μοντέλων, έχει παρατηρηθεί το ένα να παρουσιάζει μεγαλύτερο recall και το άλλο μεγαλύτερο precision. Για τον λόγο αυτό, το F1 score χρησιμοποιείται πολύ συχνά στη σύγκριση των μοντέλων, αφού συνδυάζει και τα δύο αυτά μεγέθη. Οι τιμές που λαμβάνει είναι κλίμακας [0,1]. Όσο πιο κοντά στη μονάδα είναι η τιμή του F1 score, τόσο πιο ακριβές και αξιόπιστο θεωρείται το μοντέλο. Ως ορισμός προκύπτει από το ακόλουθο κλάσμα:

$$\frac{P+R}{2(P*R)}$$

LSTM Model

```
#LSTM Model
print(classification_report(test_y.argmax(axis=1), testPredict.argmax(axis=1)))
```

```
              precision    recall  f1-score   support

           0       0.65      0.88      0.75      2988
           1       0.66      0.34      0.45      2095

    accuracy                           0.66      5083
   macro avg       0.66      0.61      0.60      5083
weighted avg       0.66      0.66      0.63      5083
```

Εικόνα 35: Classification Report για το LSTM μοντέλο

XGBoost Model

```
#XGB Model
print(classification_report(test_y_xgb,predictions))
```

```
              precision    recall  f1-score   support

         0.0       0.63      0.91      0.75      2988
         1.0       0.65      0.25      0.36      2095

    accuracy                           0.64      5083
   macro avg       0.64      0.58      0.55      5083
weighted avg       0.64      0.64      0.59      5083
```

Εικόνα 36: Classification Report για το xgboost μοντέλο

Αρχικά παρατηρούμε εύκολα ότι τα αποτελέσματα δεν διαφέρουν και πολύ μεταξύ τους. Στις πρώτες δύο γραμμές περιγράφονται οι τιμές που λαμβάνει κάθε μοντέλο ανάλογα με την κλάση. Βλέπουμε για παράδειγμα ότι όταν πρόκειται για το label = 0, το LSTM μοντέλο επιτυγχάνει ποσοστό 65% ενώ το XGBoost 63%. Αντίθετως στο recall υπερτερεί το XGB με 91% έναντι του 88% που προβλέπει το LSTM. Ωστόσο αυτό έχει ως αποτέλεσμα για την μηδενική κλάση και τα δύο μοντέλα έχουν το ίδιο F1-score (0.75). Σύμφωνα λοιπόν με τη μηδενική κλάση τα δύο μοντέλα φαίνεται να είναι εξίσου καλά.

Στην δεύτερα γραμμή, εντοπίζονται τα αντίστοιχα ευρήματα της δεύτερης κλάσης (label =1). Εδώ παρατηρούμε ότι το precision δεν έχει μεγάλη διαφορά, με 66% στο LSTM και 65% στο XGBoost, αλλά αυτή τη φορά το LSTM υπερτερεί και στο recall με ποσοστό 34% έναντι

25% του XBGoost. Έτσι προκύπτει αντίστοιχα και υψηλότερο F1 score στο LSTM μοντέλο (0.45).

Παρόλο όμως που το LSTM έχει υψηλότερη απόδοση στην κλάση 1, δεν είναι ιδιαίτερα υψηλά τα ποσοστά του. Συγκεκριμένα φαίνεται ότι δεν μπόρεσε να προβλέψει ούτε τις μισές τιμές που θα έπαιρναν τιμή 1, και αυτό αντίστοιχα είναι η αιτία του χαμηλού F1 score.
Η στήλη support δείχνει τον αριθμό που έχει εμφανιστεί η κάθε κλάση στο σύνολο των δεδομένων. Πρόκειται για μία τιμή η οποία είναι ίδια σε όλα τα μοντέλα αλλά μπορεί να χρησιμοποιηθεί βοηθητικά στη διαδικασία αξιολόγησης. Ωστόσο στον παρόν report φαίνεται ότι υπάρχουν αρκετές τιμές και από τις δύο κλάσεις, και δεν εντοπίζεται πρόβλημα ανισορροπίας δεδομένων, οπότε θα θέλαμε να επιτυγχάναμε καλύτερα αποτελέσματα για την δεύτερη κλάση (label =1). Σε περίπτωση για παράδειγμα που όλα τα δεδομένα πλην μίας τιμής ήταν στην πρώτη κλάση, θα μπορούσαμε απλά να τα προβλέπαμε όλα ως την πρώτη κλάση και θα επιτυγχάναμε υψηλή ακρίβεια, χωρίς ωστόσο να σημαίνει ότι το μοντέλο ήταν αποτελεσματικό.

5.6.3 Roc Curve και Roc Auc Curve

Τέλος εξίσου σημαντικές πληροφορίες δίνουν και οι καμπύλες Roc curve (Receiver operating characteristic) και Roc Auc curve (Area under Curve). Η χρήση της καμπύλης roc δίνει την δυνατότητα εύκολης οπτικής εκτίμησης και σύγκρισης δύο ή περισσότερων προβλεπτικών μοντέλων. Η απεικόνιση της αναπαριστά γραφικά την σχέση μεταξύ του ποσοστού TP και FP καθώς μεταβάλλεται το threshold. Τα ποσοστά TP και FP ορίζονται ως ακολούθως:

$$\text{TP rate} = \frac{TP}{TP + FN}$$

και

$$\text{FP rate} = \frac{FP}{FP + TN}$$

Στο διάγραμμα το μοντέλο του οποίου η καμπύλη τείνει να είναι πιο ψηλά θεωρείται βέλτιστο. Όσο μεγαλύτερη κυρτότητα ως προς την άνω αριστερή γωνία έχει ένα μοντέλο, τόσο καλύτερη ικανότητα πρόβλεψης έχει. Εν Αντιθέτως τα μοντέλα των οποίων οι καμπύλες βρίσκονται κοντά στη διχοτομική καμπύλη (διαγώνιο) θεώνται μη ικανά, και δεν θα πρέπει να λαμβάνονται υπόψη οι προβλέψεις τους.

Το εμβαδόν που βρίσκεται κάτω από την ROC καμπύλη ονομάζεται AUC (area under curve). Το εμβαδό αυτό εκφράζει την πιθανότητα να πραγματοποιηθεί μία σωστή ταξινόμηση. Κατ' επέκταση όσο πιο κοντά στη μονάδα είναι αυτό το εμβαδό τόσο μεγαλύτερη είναι αντίστοιχα και η πιθανότητα το μοντέλο να προβλέψει την μείωση ή την αύξηση της τιμής των σπιτιών.

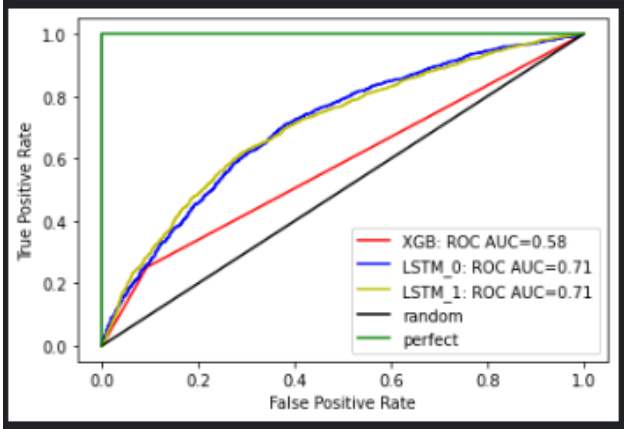
Στην παρακάτω εικόνα βλέπουμε το διάγραμμα της καμπύλης Roc για κάθε ένα από τα προβλεπτικά μοντέλα που χτίσαμε. Με πράσινο χρώμα είναι τα επιθυμητό βέλτιστο σημείο που θα θέλαμε να φτάνουμε, ενώ με μαύρο χρώμα είναι η γραμμή που θα έφερνε ένα τυχαίο αποτέλεσμα (διχοτόμος), και χρησιμοποιείται ως το worst case scenario. Με κόκκινο χρώμα είναι η roc καμπύλη του xgb μοντέλου, ενώ με μπλε και κίτρινο αντικατοπτρίζεται αντίστοιχα κάθε κλάση του LSTM μοντέλου.

Όπως είναι εύκολα αντιληπτό η ROC καμπύλη του LSTM μοντέλου είναι πολύ υψηλότερα από την καμπύλη του xgboost, συνεπάγοντας καλύτερα αποτελέσματα. Αντιστοίχως το εμβαδόν κάτω από την ROC καμπύλη (AUC) είναι μεγαλύτερο για το LSTM μοντέλο. Συγκεκριμένα το xgb μοντέλο έχει μόλις 58% πιθανότητα για σωστή πρόβλεψη ταξινόμησης, το οποίο είναι μόλις λίγο μεγαλύτερο από την τυχαία πρόβλεψη που θα είχε 50%. Το LSTM όμως φαίνεται να έχει 71% πιθανότητα, μεγαλύτερη από το xgb και άρα μπορεί να μεγαλύτερη πιθανότητα να προβλέψει την σωστή κλάση (αύξηση ή μείωση).

```
#***********  XGB Model  ***********
#ROC curve and AUC in LSTM XGB model
fpr_XGB, tpr_XGB, thresholds_XGB = roc_curve(test_y_xgb, predictions)
auc_xgb = roc_auc_score(test_y_xgb, predictions)

#********  LSTM Model  **********
#ROC curve and AUC in LSTM model
fpr_lstm = dict()
tpr_lstm = dict()
auc_lstm = dict()
for i in range(2):
    fpr_lstm[i], tpr_lstm[i] , _ = roc_curve(test_y[:, i], testPredict[:, i])
    auc_lstm[i] = auc(fpr_lstm[i], tpr_lstm[i])
```

Εικόνα 37: Δημιουργία ROC curve για κάθε μοντέλο

```
plt.plot(fpr_XGB, tpr_XGB,'r-',label = 'XGB: ROC AUC=%.2f' %(auc_xgb))
plt.plot(fpr_lstm[0],tpr_lstm[0],'b-', label= 'LSTM_0: ROC AUC=%.2f' %(auc_lstm[0]))
plt.plot(fpr_lstm[1],tpr_lstm[1],'y-', label= 'LSTM_1: ROC AUC=%.2f' %(auc_lstm[1]))
plt.plot([0,1],[0,1],'k-',label='random')
plt.plot([0,0,1,1],[0,1,1,1],'g-',label='perfect')
plt.legend()
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```



Εικόνα 38: Γράφημα της ROC curve και σύγκριση της AUC κάθε μοντέλου

6. Συμπεράσματα

6.1 Συμπεράσματα διπλωματικής εργασίας

Στο πλαίσιο της παρούσας διπλωματικής εργασίας αναλύσαμε την χρήση και εφαρμογή των αλγορίθμων μηχανικής μάθησης στα ανοιχτά κυβερνητικά δεδομένα. Συγκεκριμένα για τους σκοπούς αυτούς χρησιμοποιήθηκαν δύο διαφορετικής θεωρίας αλγόριθμοί. Αφού αναλύθηκε το θεωρητικό υπόβαθρο αυτών, στην συνέχεια περιγράφικε η εκπαίδευση και η συμπεριφορά των αντίστοιχων μοντέλων, οι οποίοι χρησιμοποιούσαν ως input data τα ΑΚΔ της Σκωτίας. Έγινε λεπτομερείς αναφορά στα πλεονεκτήματα και τα οφέλη που προέρχονται από την συλλογή και αξιοποίηση των ΑΚΔ και ως προς τους ίδιους τους οργανισμούς αλλά και ως προς τους πολίτες. Επίσης αναφερθήκαμε στον τρόπο που πρέπει να γίνεται η συλλογή και αποθήκευση τέτοιων δεδομένων έχοντας πάντα ως βασικό γνώμονα να διασφαλίζεται το αίσθημα ελευθερίας και ασφάλειας των πολιτών. Είδαμε ότι για να δώσουν οι πολίτες αληθή και ακριβή δεδομένα θα πρέπει σαφώς να νιώθουν εμπιστοσύνη στο κράτος και τους κυβερνητικούς εκπροσώπους τους, διαφορετικά έγκειται πάντα ο κίνδυνος λήψης αποφάσεων βασισμένες σε ψευδή στοιχεία.

Έτσι εφαρμόστηκε στη πράξη η θεωρία της χρήσης των ΑΚΔ δημιουργώντας δύο μοντέλα πρόβλεψης, ώστε να βρεθεί το βέλτιστο, το οποίο θα επιτύχει την καλύτερη δυνατή πρόβλεψη της μεταβολής των τιμών των σπιτιών στις μικρές γεωγραφικές περιοχές της Σκωτίας. Για τον λόγο αυτό αφού αναπτύχθηκαν τα μοντέλα έγινε η σύγκριση των αποτελεσμάτων τους. Το πρώτο μοντέλο ήταν το LSTM (Long Short Term Memory), όπου είναι ένας αλγόριθμός deep learning. Στη συνέχεια δημιουργήθηκε ένα xgboost μοντέλο. Πρόκειται στην ουσία για μία τεχνική boosting, η οποία είναι ευρέως διαδεδομένη στους διαγωνισμούς του Kaggle, καθώς φαίνεται να έχει πιο συχνά τα καλύτερα αποτελέσματα αξιολόγησης, και κατ' επέκταση να κερδίζει στους διαγωνισμούς. Η έννοια και το θεωρητικό υπόβαθρό των συγκεκριμένων αλγορίθμων έχει περιγραφή αναλυτικά σε προηγούμενο κεφάλαιο. Για τους σκοπούς αυτούς αναλύθηκε η επιλογή των κατάλληλων παραμέτρων (parameter tuning) ώστε να βελτιστοποιήσουν τα αποτελέσματα για κάθε ένα από τα δύο μοντέλα που δημιουργήθηκαν. Τέλος έγινε η αξιολόγηση και σύγκριση των δύο μοντέλων.

Σε γενικές γραμμές, μπορούμε να πούμε ότι αρχικά τα δύο μοντέλα παρουσίαζαν σχεδόν παρόμοια συμπεριφορά σύμφωνα με ορισμένους δείκτες (accuracy, precision). Ωστόσο αυτό που παρατηρείται ότι το LSTM μοντέλο φαίνεται από αποδίδει καλύτερα και σύμφωνα με

αριθμητικούς δείκτες (F1 Score) αλλά και σε οπτική απεικόνιση (ROC curve, AUC). Έτσι με την προϋπόθεση ότι οι μεταβλητές παραμένουν οι ίδιες, μπορούμε να θεωρήσουμε ότι το LSTM μοντέλο που αναπτύχθηκε ήταν σε θέση να προβλέψει καλύτερα, και πιο αποτελεσματικά την σωστή ταξινόμηση.

Συμπερασματικά, μπορούμε να αναφέρουμε ότι είναι πλέον γεγονός ότι οι τεχνολογικές εξελίξεις, οι καινοτομίες αλλά και ο ολοένα αυξανόμενος όγκος των δεδομένων που δημιουργούνται καθημερινά, έχουν επιφέρει ήδη αλλαγές, όχι μόνο τον τομέα των επιχειρήσεων, αλλά και τον τρόπο λειτουργίας και αξιολόγησης των κρατικών οργανισμών. Οι αναπτυγμένες χώρες μπορούν να επωφεληθούν άμεσα και να αναπτύξουν μία data driven decision κυβερνητική πολιτική. Αυτό πρακτικά σημαίνει ότι οι πολιτικές αποφάσεις που παίρνοντας δεν θα είναι μόνο βασισμένες στην θεωρία της πολιτικής επιστήμης ή στην εμπειρική γνώση, αλλά θα χρησιμοποιούνται όλες τις πληροφορίες που δίνονται αξιοποιώντας τα ίδια τα δεδομένα. Με σαφήνεια λοιπόν μπορούμε να καταλήξουμε στο συμπέρασμα ότι αυτή είναι η κατεύθυνση πριν ληφθεί κάποια απόφαση. Ήδη όλοι οι μεγάλοι οργανισμοί έχουν αντιληφθεί τα οφέλη που υπόκεινται, και σίγουρα προβλέπεται ακόμη μεγαλύτερη εξελιξή στο μέλλον.

6.2 Μελλοντική Έρευνα

Ολοκληρώνοντας την εργασία είναι σημαντικό να αναφερθεί ότι η ανάλυση καθώς και ο ίδιος ο σχεδιασμός των αλγορίθμων θα μπορούσαν σίγουρα να βελτιωθούν. Όπως έχει ήδη αναφερθεί ένα από τα βασικά πλεονεκτήματα όλων των RNN αλγορίθμων οπότε και του LSTM είναι ότι μπορεί να χειριστεί με ευκολία τα μεγάλα δεδομένα (big data). Αυτό συμβαίνει διότι ο ίδιος ο αλγόριθμος πραγματοποιεί feature extraction, εντοπίζοντας και λαμβάνοντας υπόψη εκείνες τις μεταβλητές που χρειάζεται να χρησιμοποιήσει περισσότερο για την καλύτερη πρόβλεψη. Έτσι προτείνεται προς μελέτη και έρευνα η εύρεση των κατάλληλων ποιοτικών μεταβλητών οι οποίες θα μπορούσαν να βελτιώσουν την πρόβλεψη του μοντέλου. Σίγουρα ένα πρώτο στάδιο βελτίωσης θα ήταν να χρησιμοποιούν επιπλέον μεταβλητές οι οποίες θα περιγράφουν τις μικρές γεωγραφικές περιοχές της Σκωτίας και να συγκριθούν τα αποτελέσματα.

Επιπροσθέτως ένα ακόμη στάδιο εξέλιξης και έρευνας θα ήταν να παραμετροποιηθεί ακόμη και ο τρόπος που το ίδιο το μοντέλο λαμβάνει τα input data. Στα πλαίσια της συγκεκριμένης εργασίας η συλλογή των δεδομένων από την ιστοσελίδα των ανοιχτών κυβερνητικών δεδομένων της Σκωτίας έγινε δημιουργώντας tables με τις ενδιαφερόμενες μεταβλητές απευθείας στο site, ενώ στη συνέχεια έγινε το διάβασμα αυτών ως pandas dataframe. Θα λοιπόν ιδιαίτερα ενδιαφέρον αν δημιουργηθεί το σύστημα που να καλεί απευθείας τις επιλεγούσες μεταβλητές, ώστε να υπάρχουν real time data, και σύμφωνα με αυτά τα τρέχει το μοντέλο. Έτσι κάθε χρόνο που θα προστίθενται νέα δεδομένα, θα πραγματοποιείται αντίστοιχα και η νέα πρόβλεψη. Επιπλέον σε περίπτωση που θα γίνει κάποια ενημέρωση ή ανανέωση των ήδη δημοσιευμένων δεδομένα, το μοντέλο θα μπορεί και πάλι να πραγματοποιεί πρόβλεψη λαμβάνοντας υπόψη κάθε καινούρια μεταβολή των στοιχείων.

7. Βιβλιογραφία

1. A. Gandomi, M. Haider (2014) Beyond the hype: Big data concepts, methods, and analytics

2. A. Purwanto, A. Zuiderwijk, M. Janssen (2020) Citizen engagement with open government data: Lessons learned from Indonesia’s presidential election

3. L.Parisi, R, Ma, N, RaviChandran, M. Lanzillotta (2021) Hyper-sinh: An accurate and reliable function from shallow to deep learning in TensorFlow and Keras

4. Zhang, Lei Xu, Xiangyu Zhang, Baowen Xu (2021) Quantifying the interpretation overhead of Python

5. N. Ketkar, J. Moolayil (2021) Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch

6. J. Sanz, M. Sesma-Sara, H. Bustince (2021) A fuzzy association rule-based classifier for imbalanced classification problems

7. Issam Dawoud, Adewale F. Lukman, Abdul-Rahaman Haadi (2021) A new biased regression estimator: Theory, simulation and application

8. PrediHanaa Salem,A.E. Kabeel, Emad M.S. El-Said, Omar M. Elzeki (2021) ictive modelling for solar power-driven hybrid desalination system using artificial neural network regression with Adam optimization

9. Diederik P. Kingma, Jimmy Lei Ba (2015) Adam: A method for stochastic optimization
Arnulf Jentzen, Philippe von Wurstemberger (2020) Lower error bounds for the stochastic gradient descent optimization algorithm: Sharp convergence rates for slowly and fast decaying learning rates

10. Zihan Chang, Yang Zhang, Wenbo Chen (2019) Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform

11. M. Murray, V. Abrol, J. Tanner (2021) Activation function design for deep networks: linearity and effective initialization

12. Luca Parisi, Daniel Neagu, Renfei Ma, Felician Campean (2021) Quantum ReLU activation for Convolutional Neural Networks to improve diagnosis of Parkinson's disease and COVID-19

13. Michael Yeung, Evis Sala, Carola-Bibiane Sch ̈onlieb, Leonardo Rundo (2021) Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation

14. Buse Melis Ozyildirim, Mariam Kiran (2021) Levenberg–Marquardt multi-classification using hinge loss function

15. Mehryar Mohri, Afshin Rostamizadeh, Ammet Talwalkar (2012) Foundation of Machine Learning

16. Damien Gruson, Thibault Helleputte, Patrick Rousseau, David Gruson (2019) Data science, artificial intelligence, and machine learning: Opportunities for laboratory medicine and the value of positive regulation

17. Jian Zhou, Yingui Qiu, Danial Jahed Armaghani b, Wengang Zhang, Chuanqi Li, Shuangli Zhu, Reza Tarinejad (2020) Predicting TBM penetration rate in hard rock condition: A comparative study among six XGB-based metaheuristic techniques

16. Auref Rostamian, Ehsan Heidaryan, Mehdi Ostadhassan (2021) Evaluation of different machine learning frameworks to predict CNL-FDC-PEF logs via hyperparameters optimization and feature selection

17. Jerome Friedman, Trevor Hastie, and Robert Tibshirani (2000) Additive logistic regression: A statistical view of boosting

18. Raffaele Pugliese, Stefano Regondi, Riccardo Marini (2021) Machine learning-based approach: Global trends, research directions, and regulatory standpoints

19. Tianqi Chen, Tong He (2014) Higgs Boson Discovery with Boosted Trees
Tianqi Chen, Carlos Guestrin (2016) XGBoost: A Scalable Tree Boosting System

20. Qinghua Gu, Yinxin Chang, Naixue Xiong, Lu Chen (2020) Forecasting Nickel futures price based on the empirical wavelet transform and gradient boosting decision trees

21. Wentao Cai, Ruihua Wei, Lihong Xu, Xiaotao Ding (2020) A method for modeling greenhouse temperature using gradient boost decision tree

22. Thibaut Vaulet, Maya Al-Memar, Hanine Fourie, Shabnam Bobdiwala, Srdjan Saso, Maria Pipi, Catriona Stalder, Phillip Bennett, Dirk Timmerman, Tom Bourne, Bart De Moor (2021) Gradient boosted trees with individual explanations: An alternative to logistic regression for viability prediction in the first trimester of pregnancy

23. Surjeet Kumar Yadav, Saurabh Pal (2012) Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification

24. Keiran Hardy, Alana Maurushat (2017) Opening up government data for Big Data analysis and public benefit

25. Jamie Barryhill, Kevin Kok Heang, Rob Clogher, Keegan McBride (2019) Hello, Word: Artificial intelligence and its use in the public sector

26. Rouzbeh Meymandpour, Joseph G.Davis (2015) A semantic similarity measure for linked data: An information content-based approach

27. Hongming Cai, Athanasios V. Vasilakos (2017) Managing the Web of Things

Zongben Xu, Niansheng Tang, Chen Xu, Xueqi Cheng (2021) Data science: connotation, methods, technologies, and development

28. Victoria Wang David Shepherd (2020) Exploring the extent of openness of open government data – A critique of open government datasets in the UK

29. Rachel Gong and Hui San Chiam (2019) Personal Data Privacy and Surveillance Capitalism

30. LuciaNalbandian (2022) Increasing the Accountability of Automated Decision-Making Systems: An Assessment of the Automated Decision-Making System Introduced in Canada's Temporary Resident Visa Immigration Stream

31. Antonio Vetrò, Lorenzo Canova, Marco Torchiano, Camilo Orozco Minotas, Raimondo Iemma, Federico Morando (2016) Open data quality measurement framework: Definition and application to Open Government Data

32. Archon Fung, Hollie Russon Gilman, Jennifer Shkabatur (2015) Technology for Democracy in Development: Lessons from Seven Case Studies

33. Katharina Sielemann, Alenka Hafner, Boas Pucker (2020) The reuse of public datasets in the life sciences: potential risks and rewards

34. Anders Haug, Frederik Zachariassen, Dennis van Liempd (2010) The costs of poor data quality

35. Barbara Ubaldi (2013) Open Government Data, Towards Empirical Analysis of Open Government Data Initiatives

36. Yarin Gal Zoubin Ghahramani (2016) A Theoretically Grounded Application of Dropout in Recurrent Neural Networks

37. Karsoliya S. (2012) Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture

38. H. Sak, A. Senior, F. Beaufays (2014) Long Short-Term Memory based deep recurrent neural networks for Large scale acoustic modeling

39. A. Graves, M. Liwicki, Sa. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber (2009) A Novel Connectionist System for Unconstrained Handwriting Recognition

40. Graves A (2012) Supervised Sequence Labelling with Recurrent Neural Networks
Tim G. Davies, Zainab Ashraf Bawa (2012) The Promises and Perils of Open Government Data (OGD)

41. Ian Goodfellow, Yoshua Bengio, Aaron Courville (2016) Deep Learning (Adaptive Computation and Machine Learning series)

42. Yu H., Robinson D. (2012) The New Ambiguity of "Open Government"

43. Yoshua Bengio (2012) Deep Learning of Representations for Unsupervised and Transfer Learning

44. Galih Salman, Yaya Heryadi, Edi Adburahman, Wayan Suparta (2018) Single Layer & Multi - layer Lon Short - Term Memory (LSTM) model with Intermediate variables for Weather Forecasting

45. Leonida Mutuku, Christine Mahihu (2014) Open data in developing countries: Understanding the impacts of Kenya open data applications and services

46. Bovens, M. (2005) The Concept of Public Accountability

47. Josefin Lassinantti (2019) Re - Use of public sector open data

48. Maguire, S. (2011) Can Data Deliver Better Government?

8. Παράρτημα

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
from scipy import stats
import random
import re
import math
from sklearn.preprocessing import MinMaxScaler
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers.core import Dropout
from keras.layers import Dense
from keras.callbacks import EarlyStopping
from keras.utils.vis_utils import plot_model
from sklearn.metrics import mean_squared_error
from sklearn.metrics import explained_variance_score
from sklearn.metrics import r2_score
from sklearn.model_selection import GridSearchCV
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from xgboost import plot_importance
import xgboost as xgb
from sklearn.metrics import classification_report
import numpy as np
from scipy import interp
import matplotlib.pyplot as plt
```

```
from itertools import cycle
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

#function to stack the dataframe
def correct_dataset(df,start_year,feature,geo='Zone'):
    df.loc[:,'Feature Identifier']=df.loc[:,'Feature Identifier'].apply(lambda x: x.replace('http://statistics.gov.scot/id/statistical-geography/',''))
    n_columns = len(df.columns)
    name_columns = ['Code_'+geo,'Name_'+geo]
    s = int(start_year)-1
    for i in df.columns:
        if 'Feature' not in i:
            s = s+1
            str_year = str(s)
            name_columns.append(str_year)
    df.columns = name_columns
    df = df.set_index(['Code_'+geo,'Name_'+geo],append=True).stack(dropna=False).reset_index().drop('level_0',1)
    correct_name_cols = ['Code_'+geo,'Name_'+geo,'Year']
    correct_name_cols.append(feature)
    df.columns = correct_name_cols
    return df

#handling missing data, with interpolatelinear
#set year, as date, and then set indexes
def missing_data(df,geo='Zone'):
```

```
    col =df.columns.drop(['Code_'+geo,'Name_'+geo,'Year'])
    #col.append(colname)
    df.loc[:,'Year'] = pd.to_datetime(df.loc[:,'Year'], format='%Y')
    df = df.set_index('Year')
    df_interpol = df.groupby(['Code_'+geo,'Name_'+geo])\
            .resample('Y')\
            .mean()
    df_interpol[col]=df_interpol[col].interpolate()
    df_interpol = df_interpol.reset_index()
    df_interpol.loc[:,'Year'] = df_interpol.loc[:,'Year'].dt.year.astype(str)
    return df_interpol
```

#downloaded a slice for data, with count births and deaths for female and male in every 2011datazone

```
births_deths=pd.read_csv('https://statistics.gov.scot/carts/download/db0a5eb0-3aad-44fd-85a6-08ef4ee02b91?format=csv')
#create the column for the year 2018 for deaths
#by calculation the mean from the past years
births_deths['Deaths_female_2018'] = births_deths.iloc[:, [12,13,14,15]].mean(axis=1)
births_deths['Deaths_male_2018'] = births_deths.iloc[:, [16,17,18,19]].mean(axis=1)
#re order the dataframe.
order_cols =births_deths.iloc[:,:16].columns.tolist() + ['Deaths_female_2018'] +births_deths.iloc[:,16:20].columns.tolist() +['Deaths_male_2018']
births_deths = births_deths[order_cols]

#apply the correst_dataset function for births_deths dataset
a , b = 2, 7
```

```
for i in range(4):
    if i>0:
        a = a+5
        b = b+5
    temp = births_deths.iloc[:,np.r_[0,1,a:b]].copy()
    str_name =re.findall('^(.+):.*Gender = (\S+);.+',temp.columns[2])[0]
    feature = str_name[0]+'_'+str_name[1][0]
    year=int(re.findall('^.+ Reference Period = (\d\d\d\d).+',temp.columns[2])[0])
    df = correct_dataset(temp,year,feature)
    if i==0:
        df_all = df
    else:
        df_all = df_all.merge(df,how='left',on=['Code_Zone','Year','Name_Zone'])


#download the population dataset, and calculate the ratio on births and deaths.
#apply correct_dataset function, and merge all together (df_all_)
population = pd.read_csv('https://statistics.gov.scot/carts/download/1bafc10a-098c-4f03-
b0da-edcb1fa0c0e9?format=csv')
population= correct_dataset(population,2014,'Polulation')
df_all = df_all.merge(population,how='left',on=['Code_Zone','Year','Name_Zone'])

#calculate the ratio for each columns
for j in range(3,7):
    new_col = df_all.columns[j]+'_ratio'
    df_all[new_col]=np.nan
    c=5+j
    for i in range(df_all.shape[0]):
        if df_all.iloc[i,7]!=0:
```

```
        df_all.iloc[i,c]=round((float(df_all.iloc[i,j])/float(df_all.iloc[i,7]))*1000,2)

df_all.head()

births_f = df_all.iloc[:,np.r_[0,1,2,8]]
births_m = df_all.iloc[:,np.r_[0,1,2,9]]
deaths_f = df_all.iloc[:,np.r_[0,1,2,10]]
deaths_m = df_all.iloc[:,np.r_[0,1,2,11]]
del df_all

#pay_gap, not_qualified people for council areas
pay_Nqual=pd.read_csv('https://statistics.gov.scot/carts/download/c2564489-a26f-496f-ae93-ed8b8e387ffb?format=csv')
other=pd.read_csv('https://statistics.gov.scot/carts/download/0cf52ce6-0129-4a95-8729-447c2509ee02?format=csv')

paygap_full=pay_Nqual.iloc[:,np.r_[0:9]]
paygap_part=pay_Nqual.iloc[:,np.r_[0,1,9:16]]
no_qual_f = pay_Nqual.iloc[:,np.r_[0,1,16:23]]
no_qual_m = pay_Nqual.iloc[:,np.r_[0,1,23:30]]
del pay_Nqual

edu=other.iloc[:,0:8]
ante_natal_former= other.iloc[:,np.r_[0,1,8:14]]
ante_natal_never= other.iloc[:,np.r_[0,1,14:20]]
ante_natal_notknow= other.iloc[:,np.r_[0,1,20:26]]
#fire = other.iloc[:,np.r_[0,1,26:31]]
del other
```

```
#in every dataset we want year 2014-2018, extract more year if exist
# to help handling missing values function
#apply function, and then keep only years 2014-2018

edu=correct_dataset(edu,2013,"Educational attainment of school leavers")
ante_natal_former=correct_dataset(ante_natal_former,2013,"Ante-Natal Former Smoker")
ante_natal_never=correct_dataset(ante_natal_never,2013,"Ante-Natal Never Smoked")
ante_natal_notknow=correct_dataset(ante_natal_notknow,2013,"Ante-Natal Not Known")
#fire=correct_dataset(fire,2014,"Fire Not Accidental")

#a list of all dataframe for DataZone
list_df_zones =
[births_f,births_m,deaths_f,deaths_m,edu,ante_natal_former,ante_natal_never,ante_natal_n
otknow]#,fire]

#missing_data function to dataframe with missing values
list_data_with_null=[]
data_ready=[]
for data in list_df_zones:
    if (data.isnull().any()).any():
        list_data_with_null.append(data)
    else:
        data_ready.append(data)
    print(data.isnull().sum())

#the data_ready list is empty. so we use only the list_data_with_null
z=0

for i in range(len(list_data_with_null)):
```

```
    data = list_data_with_null[i].copy()
    df = missing_data(data)
    z=z+1
    if z==1:
        dfs_zone=df
    else:
        dfs_zone = dfs_zone.merge(df,how='left',on=['Code_Zone','Year','Name_Zone'])

dfs_zone.head()

print(dfs_zone.isnull().sum())
print(dfs_zone.describe())


paygap_full=correct_dataset(paygap_full,2013,"Gender PayGap full_time",'Area')
paygap_part=correct_dataset(paygap_part,2013,"Gender PayGap part_time",'Area')
no_qual_f=correct_dataset(no_qual_f,2013,"Female Adults with No qualifications",'Area')
no_qual_m=correct_dataset(no_qual_m,2013,"Male Adults with No qualifications",'Area')

#list of all datasets for coucnil areas
list_df_areas=[paygap_full,paygap_part,no_qual_f,no_qual_m]

#missing_data function to dataframe with missing values

list_data_with_null=[]
data_ready=[]
for data in list_df_areas:
    if (data.isnull().any()).any():
        list_data_with_null.append(data)
```

```
    else:
        data_ready.append(data)
    print(data.isnull().sum())


count=0
for i in range(len(list_data_with_null)):
    count+=1
    data = list_data_with_null[i].copy()
    df = missing_data(data,'Area')
    if count==1:
        dfs_area =df
    else:
        dfs_area = dfs_area.merge(df,how='left',on=['Code_Area','Year','Name_Area'])

print(dfs_area.isnull().sum())

paygap_full=correct_dataset(paygap_full,2013,"Gender PayGap full_time",'Area')
paygap_part=correct_dataset(paygap_part,2013,"Gender PayGap part_time",'Area')
no_qual_f=correct_dataset(no_qual_f,2013,"Female Adults with No qualifications",'Area')
no_qual_m=correct_dataset(no_qual_m,2013,"Male Adults with No qualifications",'Area')

#list of all datasets for coucnil areas
list_df_areas=[paygap_full,paygap_part,no_qual_f,no_qual_m]

#missing_data function to dataframe with missing values

list_data_with_null=[]
data_ready=[]
```

```
for data in list_df_areas:
    if (data.isnull().any()).any():
        list_data_with_null.append(data)
    else:
        data_ready.append(data)
    print(data.isnull().sum())



count=0
for i in range(len(list_data_with_null)):
    count+=1
    data = list_data_with_null[i].copy()
    df = missing_data(data,'Area')
    if count==1:
        dfs_area =df
    else:
        dfs_area = dfs_area.merge(df,how='left',on=['Code_Area','Year','Name_Area'])


print(dfs_area.isnull().sum())


#read house prices in small areas, rename the columns, and make proper the Identifier to match the lookup dataset
prices = pd.read_csv('https://statistics.gov.scot/carts/download/e9405718-8dd0-41a4-baeb-ba525fa4e973?format=csv')
prices['Feature Identifier'] = prices['Feature Identifier'].str.replace('http://statistics.gov.scot/id/statistical-geography/','')
prices.columns = ['Code_Zone','Name_Zone','2014','2015','2016','2017','2018']
```

```
#dataset to match the 2011Data Zoen with the Council Area they belong
lookup=pd.read_csv('https://statistics.gov.scot/downloads/file?id=1ab6565e-10e0-4888-b91c-4ae6821b30d7%2FDatazone2011lookup+%288%29.csv')
#create a dataframe with council areas ans DZone2011 to match it
DZ_CA = lookup[['DZ2011_Code','DZ2011_Name','LA_Code','LA_Name']]


#now the dataZone has the code of Council Areas they belong
prices = prices.merge(DZ_CA,how='left',left_on='Code_Zone',right_on='DZ2011_Code')
prices=prices.drop(['DZ2011_Code','DZ2011_Name'],axis=1)

df_prices = prices.set_index(['Code_Zone','Name_Zone','LA_Code','LA_Name'],append=True).stack().reset_index().rename(
columns={'level_5':'Year',0:'Price'}).drop('level_0',1)

data = df_prices.merge(dfs_zone,how='left',on=['Code_Zone','Year','Name_Zone'])
data = data.merge(dfs_area,how='left',left_on=['LA_Code','Year'],right_on=['Code_Area','Year'])
data=data.drop(['Code_Area','Name_Area'],axis=1)


del df_prices, dfs_zone, dfs_area

data.head()

data.describe()

aberdeen = random.sample(DZ_CA[DZ_CA['LA_Name']=='Aberdeen City'].iloc[:,0].tolist(),5)
```

```
striling = random.sample(DZ_CA[DZ_CA['LA_Name']=='Stirling'].iloc[:,0].tolist(),5)
fig, ax = plt.subplots(figsize=(10, 5))
label=[]
for i in range(0,5):
    y=striling[i]
    dataplot = data[data['Code_Zone']==y][['Year','Price']]
    dataplot.sort_values(by=['Year'], inplace=True)
    label.append(data[data['Code_Zone']==y].iloc[0,1])
    ax.plot(dataplot.Year,dataplot.Price)
    ax.legend(label, title='title', bbox_to_anchor=(1.05, 1), loc='upper left')
    x_order=dataplot['Year'].sort_values().to_list()
    ax.set_xticklabels(x_order)
    ax.set_title("Prices per Year")
    ax.set_xlabel('Year')
    ax.set_ylabel('Price')

fig, ax = plt.subplots(figsize=(10, 5))
label=[]
for i in range(0,5):
    y=aberdeen[i]
    dataplot = data[data['Code_Zone']==y][['Year','Price']]
    dataplot.sort_values(by=['Year'], inplace=True)
    label.append(data[data['Code_Zone']==y].iloc[0,1])
    ax.plot(dataplot.Year,dataplot.Price)
    ax.legend(label, title='title', bbox_to_anchor=(1.05, 1), loc='upper left')
    x_order=dataplot['Year'].sort_values().to_list()
    ax.set_xticklabels(x_order)
    ax.set_title("Prices per Year")
    ax.set_xlabel('Year')
```

```
    ax.set_ylabel('Price')

data_final=data
data_final.sort_values(by=['Code_Zone','Year'], inplace=True)
data_final['Price-1']=data_final.groupby('Code_Zone')['Price'].shift(-1)
data_final['Price-2']=data_final.groupby('Code_Zone')['Price'].shift(-2)
data_final=data_final.dropna()
#if we do this, we have to drop the first two rows from each area
#data_final=data_final.iloc[2:,:]

data_final.head()

#dataset to categorical problem
data_final_to_categ = data_final

#dataset to categorical problem
data_final_to_categ = data_final

y_list = list()

for i in range(data_final.shape[0]):
    if math.isnan(data_final['Price-1'].iloc[i]):
        y=np.nan
    else:
        y=1 if data_final['Price'].iloc[i]-data_final['Price-1'].iloc[i]>=0 else 0
    y_list.append(y)

data_final['y']=y_list
```

```
cols_o=['y']+ cols_x
data_final = data_final[cols_o]
data_final.head()

#start building the model
#first create the traind and test sample
#convert the final dataframe into values

data_final_model=data_final.values

type(data_final_model)
data_final_model = data_final_model.astype('float32')

# normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(data_final_model)

#first step split data into train and test. We decide to split 70/30
train_size = int(len(data_final)*.70)
train  = scaled[0:train_size,:]
test=scaled[train_size:,:]

# split into input and outputs (x variables, y variables)
train_X, train_y = train[:, 1:], train[:,0]
test_X, test_y = test[:, 1:], test[:, 0]

train_y = to_categorical(train_y)
test_y = to_categorical(test_y)
```

```
#convert numeric variables (y) to categorical

# reshape input to be 3D [samples, timesteps, features]
train_X_lstm = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X_lstm= test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)

n_outputs=train_y.shape[1]
#n_outputs=1
nodes = int(2/3*(train_X_lstm.shape[1]*train_X_lstm.shape[2]))
print('The number of hidden nodes is %.2f' %  (nodes))

model_lstm = Sequential()
model_lstm.add(LSTM(nodes, input_shape=(train_X_lstm.shape[1], train_X_lstm.shape[2])))  #input_shape=(n_feutures,look_back)
model_lstm.add(Dropout(0.2))
model_lstm.add(Dense(100, activation='softmax'))
model_lstm.add(Dense(100, activation='softmax'))
model_lstm.add(Dense(n_outputs, activation='sigmoid'))
model_lstm.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

es = EarlyStopping(monitor='val_loss', mode='min', verbose=2, patience=3)
history = model_lstm.fit(train_X_lstm, train_y, batch_size=5, validation_data=(test_X_lstm, test_y), epochs=50,callbacks=[es], verbose=2,  shuffle=False)

model_lstm.summary()

plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
```

```
plt.legend()
plt.show()

plot_model(model_lstm, to_file='lstm_model_summary.png', show_shapes=True)

plt.figure(figsize=(9,7))
plt.title('Accuracy score')
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['accuracy', 'val_accuracy'])
plt.show()

trainPredict = model_lstm.predict(train_X_lstm)
testPredict = model_lstm.predict(test_X_lstm)

score, accuracy_lstm = model_lstm.evaluate(test_X_lstm,test_y, verbose=0)
print('Test score: %.2f%%' % (score*100))
print('Test accuracy: %.2f%%' % (accuracy_lstm*100))

#XGB model, train data and evaluate

train_X_xgb, train_y_xgb = train[:, 1:], train[:,0]
test_X_xgb, test_y_xgb = test[:, 1:], test[:, 0]

model_x = XGBClassifier(objective='binary:logistic', booster='gbtree', learning_rate=0.01,
eval_metric='mlogloss', use_label_encoder=False)
max_depth = range(3,5, 7)
print(max_depth)
param_grid = dict(max_depth=max_depth)
```

```
#kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=7)
grid_search = GridSearchCV(model_x, param_grid, scoring="accuracy", cv=10, verbose=1)
grid_result = grid_search.fit(train_X_xgb, train_y_xgb)
# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

model_xgb = XGBClassifier(objective='binary:logistic', booster='gbtree', learning_rate=0.01, max_depth=3, eval_metric='mlogloss',  use_label_encoder=False)
model_xgb.fit(train_X_xgb, train_y_xgb)

model_xgb.get_booster().feature_names=cols_x

fig, ax = plt.subplots(figsize=(30,30))
xgb.plot_tree(model_xgb, num_trees=model_xgb.best_iteration, ax=ax)
plt.show()

# make predictions for test data
y_pred = model_xgb.predict(test_X_xgb)
predictions = [round(value) for value in y_pred]

accuracy_xgb = accuracy_score(test_y_xgb, predictions)
print("\nAccuracy XGB model: %.2f%%" % (accuracy_xgb * 100.0))

plot_importance(model_xgb)
plt.show()

model_lstm.evaluate(test_X_lstm, test_y, verbose=0)[1]

#****  LSTM Model  *******
```

```
#Prediction in LSTM model
trainPredict = model_lstm.predict(train_X_lstm)
testPredict = model_lstm.predict(test_X_lstm)
#Accuracy in LSTM Model
accuracy_lstm = model_lstm.evaluate(test_X_lstm, test_y, verbose=0)[1]

#****  XGB Model  *******
#Prediction in XGBmodel
y_pred = model_xgb.predict(test_X_xgb)
predictions = [round(value) for value in y_pred]
#Accuracy in LSTM Model
accuracy_xgb = accuracy_score(test_y_xgb, predictions)

print('\nAccuracy LSTM model: %.2f%%' % (accuracy_lstm*100.0))

print("\nAccuracy XGB model: %.2f%%" % (accuracy_xgb * 100.0))

#XGB Model
print(classification_report(test_y_xgb,predictions))

#LSTM Model
print(classification_report(test_y.argmax(axis=1), testPredict.argmax(axis=1)))

#* * * * * * * * * * *  XGB Model  * * * * * * * * * * *
#ROC curve and AUC in LSTM XGB model
fpr_XGB, tpr_XGB, thresholds_XGB = roc_curve(test_y_xgb, predictions)
auc_xgb = roc_auc_score(test_y_xgb, predictions)

#* * * * * * * *  LSTM Model  * * * * * * * * * *
```

```
#ROC curve and AUC in LSTM model
fpr_lstm = dict()
tpr_lstm = dict()
auc_lstm = dict()
for i in range(2):
    fpr_lstm[i], tpr_lstm[i] , _ = roc_curve(test_y[:, i], testPredict[:, i])
    auc_lstm[i] = auc(fpr_lstm[i], tpr_lstm[i])

plt.plot(fpr_XGB, tpr_XGB,'r-',label = 'XGB: ROC AUC=%.2f' %(auc_xgb))
plt.plot(fpr_lstm[0],tpr_lstm[0],'b-', label = 'LSTM_0: ROC AUC=%.2f' %( auc_lstm[0]) )
plt.plot(fpr_lstm[1],tpr_lstm[1],'y-', label = 'LSTM_1: ROC AUC=%.2f' %( auc_lstm[1]) )
plt.plot([0,1],[0,1],'k-',label='random')
plt.plot([0,0,1,1],[0,1,1,1],'g-',label='perfect')
plt.legend()
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```