Data Science - Web Mining
The case of flight passengers prediction

Γεώργιος Χείρμπος 3130230

Χειμερινό εξάμηνο 2018-2019

Contents

1 Εισαγωγή

Λόγω του μεγάλου όγκου δεδομένων που υπάρχουν διαθέσιμα στον πλανήτη, η επιστήμη των δεδομένων (Data Science) έχει προσφέρει λύσεις σε προβλήματα που θα χρειαζόταν αρκετός χρόνος και ανθρώπινοι πόροι για να αντιμετωπιστούν.

Ένα από αυτά τα προβλήματα είναι η πρόβλεψη αριθμού επιβατών για αεροπορικές πτήσεις. Μια αεροπορική εταιρία είναι αδύνατο να θεωρήσει δεδομένο αριθμό θέσεων σαν πρότυπο για την διεκπαιραίωση δρομολογίων, ή να προγραμματίσει αυθαίρετα δρομολόγια χωρίς να λάβει υπόψη τις συνήθειες μετακίνησης. Για παράδειγμα, δεν εχουν όλοι οι προορισμοί την ίδια συχνότητα επίσκεψης ή να δεσμεύσει λανθασμένα αεροπλάνα με αποτέλεσμα να υπάρχουν πολλοί κενές θέσεις ή να χρειαστεί παραπάνω θέσεις από ότι υπολόγιζε.

Καλούμαστε σε αυτόν τον διαγωνισμό, να κατασκευάσουμε ένα μοντέλο πρόβλεψης κατηγορίας πτήσης, που συμβολίζει ένα εύρος αριθμού επιβατών, με βάση δεδομένα που μας δίνονται και έπειτα να υποβάλουμε το μοντέλο μας για πρόβλεψη αγνώστων περιπτώσεων για τον έλεγχο απόδοσης του.

Έχουμε να αντιμετωπίσουμε ένα πολύ συχνό τύπο προβλήματος στην επιστήμη δεδομένων, ένα classification πρόβλημα. Το γεγονός αυτό μας διευκολήνει ως προς την χρήση κατάλληλων εργαλείων αντιμετώπισης του, τα οποία αναλύονται στις παρακάτω ενότητες.

2 Δεδομένα

2.1 Περιγραφή

Έχουμε στη διάθεση μας 8899 εγγραφές για γνωστά δεδομένα (training set) και 2229 εγγραφές για τις οποίες θα προβλέψουμε την κατηγορία.

N/N	Χαρακτηριστικα δεδομένων
1.	Ημερομηνία αναχώρισης πτήσης
2./3.	Αεροδρόμιο Αναχώρισης/Πόλη αναχώρισης
4./5.	Γεωγραφικές συντεταγμένες αεροδρομίου αναχώρισης
6./7.	Αεροδρόμιο αφιξης/Πόλη άφιξης
8./9.	Γεωγραφικές συντεταγμένες αεροδρομίου άφιξης
10.	Μέσος όρος εβδομάδων κράτησης πριν την πτήση
11.	Τυπική απόκλιση του άνω μέσου όρου
12.	Κατηγορία πτήσης (Μόνο για το training set)

2.2 Δομή

Η ημερομηνία πτήσης αναπαριστάται ως ένα μεμονομένο string της μορφής YYYY-MM-DD. Σε συντομογραφία 3 γραμμάτων έχουμε την ονομασία των αεροδρομίων αναχώρισης και άφιξης ενώ ως πλήρες όνομα έχουμε την ονομασία των πόλεων αναχώρισης και άφιξης. Οι γεωγραφικές συντεταγμένες αναπαριστούνται ως float αριθμοί με μέτρια ακρίβεια δεκαδικού (3-6). Για τους μέσους όρους και την τυπική απόκλιση έχουμε πάλι αναπαράσταση σε float αριθμούς με διακύμανση του εύρούς ακρίβειας (1-10) δεκαδικά. Η κατηγορία δίνεται σε ακέραιο αριθμο και οι τιμές είναι δεδομένες απο 0 ως 7 (8 κατηγορίες).

3 Ανάλυση και οπτικοποίση

Για να κάνουμε τα πρώτα βήματα για την αντιμετώπισης του προβλήματος είναι εύλογο να λάβουμε υπόψην μας τη φύση του και τις μεταβλητές που το επηρρεάζουν. Μια σωστή σκέψη είναι ότι βασικό

ρόλο παίζει η ημερομηνία. Η χρονική περίοδος επηρρεάζει τον τόπο επίσκεψης, συχνότητα πτήσεων (πχ. περίοδος με μεγάλη κακοκαιρία σημαίνει αραιές πτήσεις ή πολύ μικρού αριθμού επιβατών κλπ).

Με την βοήθεια της αυτοματοποιημένης ανάλυσης της πλατφόρμας Kaggle, έχουμε μια πρώτη εικόνα για τα δεδομένα.

3.1 Αρχική εικόνα

Για τις ημερομηνίες δεν είναι ξεκάθαρη η πραγματική κατάσταση καθώς υπάρχει τυχαία ομαδοποίση των τιμών το οποίο προβάλεται ως ομοιόμορφη κατανομή. Χρειάζεται διερεύνηση. Για τις τοποθεσίες (αναχώρισης και άφιξης) βλέπουμε τις δυο πρώτες συχνότερες πόλεις-αεροδρόμια. Προφανής παρατήρηση είναι οτι αυτές οι θέσεις έχουν συχνότητα ¿20% σε σύνολο ∼20 μοναδικών τιμών.

Για τις συντεταγμένες, οπως παρουσιάζονται δεν είναι δυνατό να βγάλουμε κάποιο προφανές συμπέρασμα, καθώς κατα ζεύγος (lat-long) έχουμε μεγαλύτερο νόημα. Μια ερμηνεία είναι αν υπάρχει μεγάλη συχνότερα βόρεια-νότια (long) ή ανατολικά-δυτικά (lat). Φαίνεται πως έχουμε 3 διαφορετικούς τρόπους αναπαραστασης των τοποθεσιών. Αυτό γιατί υπάρχει αντιστοίχηση [Αεροδρόμιο-Πόλη-(Lat-Long)]. Όμως στην περίπτωση μας μια πόλη έχει 2 αεροδρόμια το οποίο μας εξαναγκάζει να προσεγγίσουμε αυτές τις μεταβλητές διαφορετικά.

Ο μέσος όρος εβδομάδων κράτησης πριν την πτήση και η τυπική απόκλιση ακολουθουν προσεγγίζουν κανονική κατανομή. Από κατανομή των γνωστών κατηγοριών πτήσεων μας παρατηρούμε ότι υπάρχει διαφορετική συχνότητα για την καθε μία. Κάποιες κατηγορίες είναι αρκετά συχνές ενώ άλλες πραγματοποιούνται σε πολύ μικρό βαθμό. Ειδικα για τις κατηγορίες 1,2 και 4 έχουμε πολύ λίγες εγγραφές.

3.2 Γραφήματα

Χρειαζόμαστε περαιτέρω ανάλυση κάποιων δεδομένων για τα οποία δεν έχουμε κάποια προφανή εικόνα. Θα ξεκινήσουμε με τις ημερομηνίες.

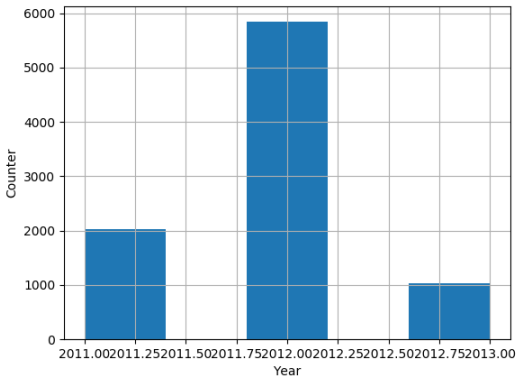

Figure 1: Κατανομή πτήσεων ανα έτος

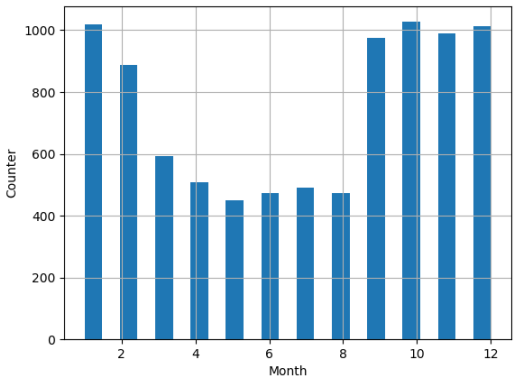

Figure 2: Κατανομή πτήσεων ανα μήνα

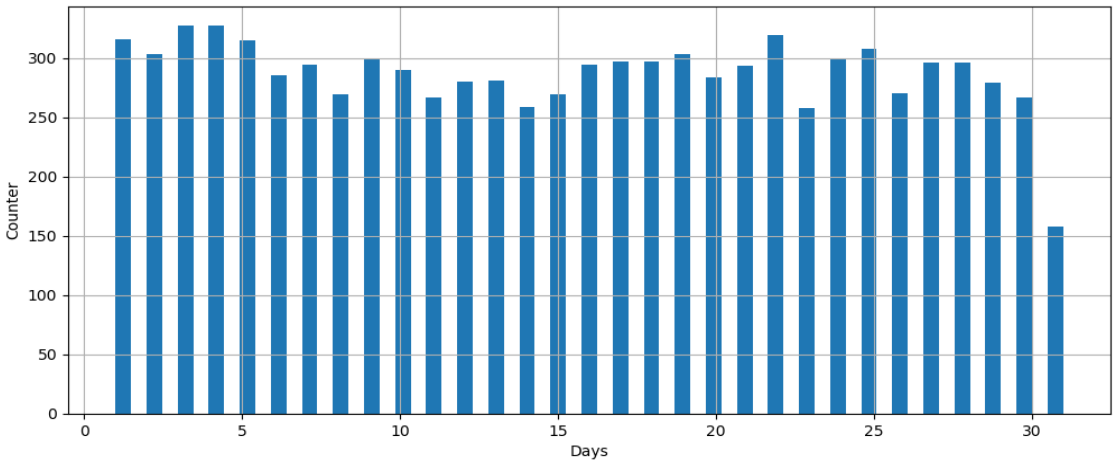

Figure 3: Κατανομή πτήσεων ανα μέρα του μήνα

Αμέσως φαίνεται μία πιο καθαρή εικόνα για την περίπτωση των ημερομηνιών. Στην περίπτωση των ετών έχουμε περίπου τα 2/3 των πτήσεων το 2012. Η υπόθεση για την χρονική περίοδο των μηνών επαληθεύεται καθώς από τον 2ο ως τον 8 μήνα υπάρχει λιγότερη μετακίνηση ενώ για της μερες προσεγγίζει αδύναμα την ομοιομορφία ενώ έχουμε σαν μέσο όρο διαφοράς έχουμε περίπου +-20 μεταξύ τους. Η 31η μέρα εχει λίγες πτήσεις αφου δεν υπάρχει σε όλους του μήνες. Παρακάτω παρουσιάζονται ομαδοποιημένες οι κατηγορίες ανα έτος μήνα μέρα.

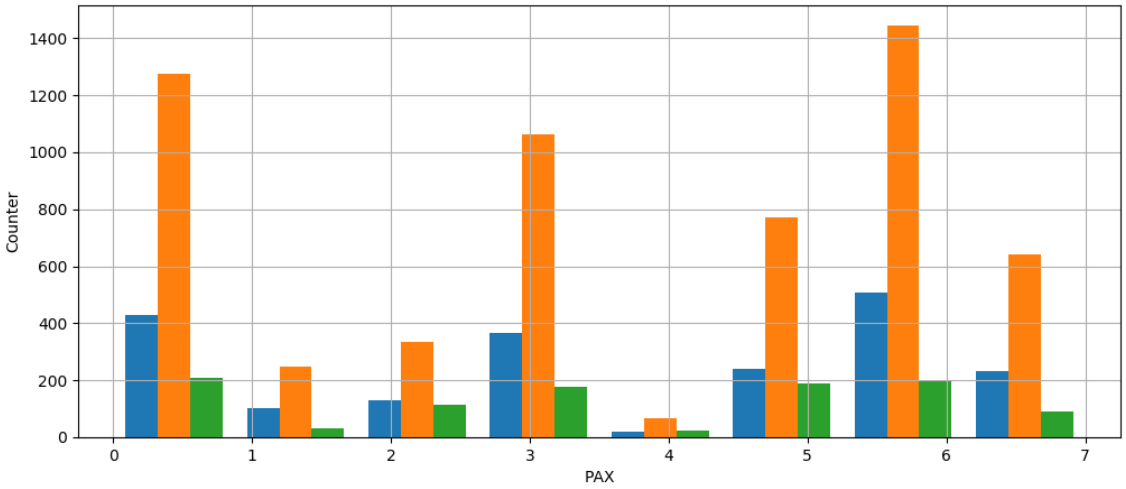

Figure 4: Κατανομή αριθμού πτήσεων ανα έτος ανα κατηγορία

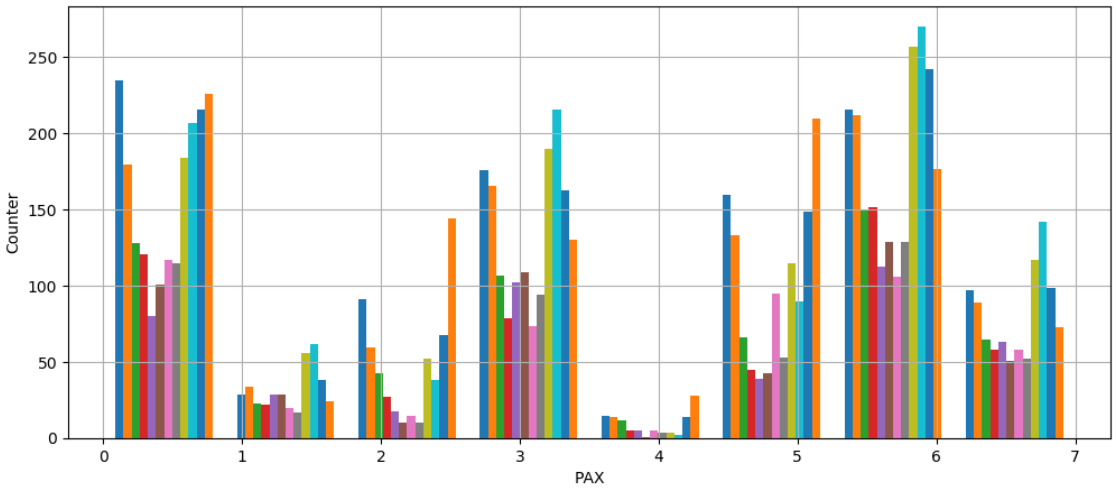

Figure 5: Κατανομή αριθμού πτήσεων ανα μήνα ανα κατηγορία

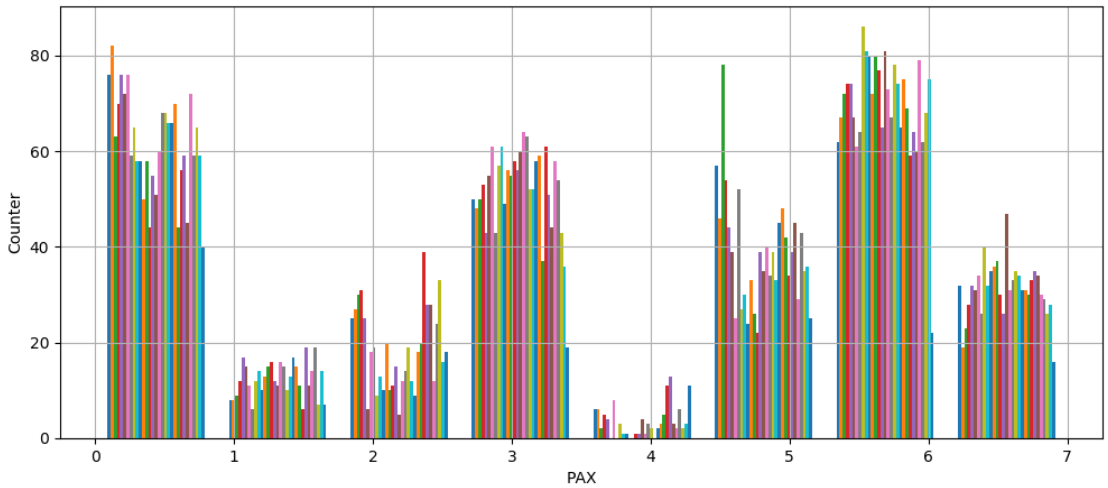

Figure 6: Κατανομή αριθμού πτήσεων ανα μέρα του μήνα ανα κατηγορία

Ο αριθμός πτήσεων ανα κατηγορία κατανεμημένη κατά έτος (Figure 4) φαίνεται να προσεγγίζει την αρχική κατανομή αριθμού πτήσεων ανα έτος (Figure 1) όμως έχουμε μια πιο αναλυτική εικόνα. Μεγαλύτερη κατανόηση για τα δεδομένα έχουμε για τις κατανομές στα Figure 5,6 όπου βλέπουμε διαφοροποιήσεις στην κατηρορία και αριθμό πτήσεων που πραγματοποιούνται ανα μήνα και ανα ημέρα του αντίστοιχου μήνα. Τέλος παρακάτω βλέπουμε τις κατανομές των πτήσεων ανά μέρα της βδομαδας.

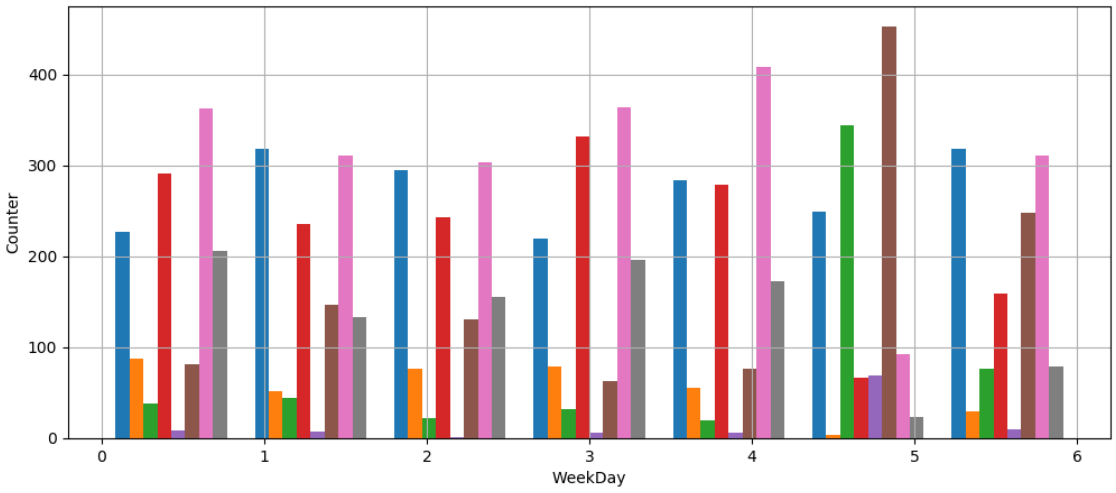

Figure 7: Κατανομή αριθμού πτήσεων ανα κατηγορία ανα μέρα βδομάδας

Μπορούμε εύκολα να ξεχωρίσουμε ότι η μέρα της βδομάδας επηρρεάζει αρκετα την κατηγορία πτήσης που θα πραγματοποιηθεί.

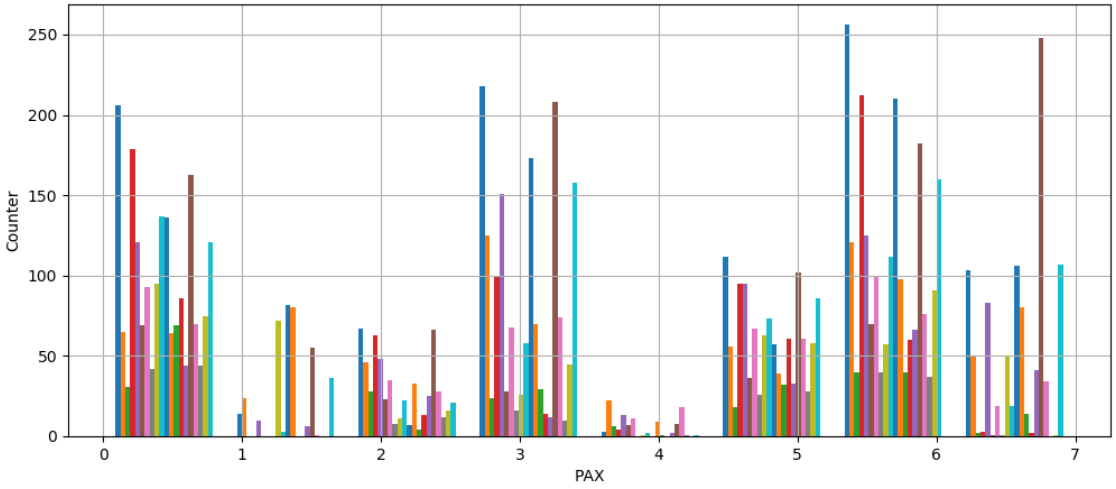

Figure 8: Κατανομή αριθμού πτήσεων ανα σταθμό αναχώρισης ανα κατηγορία

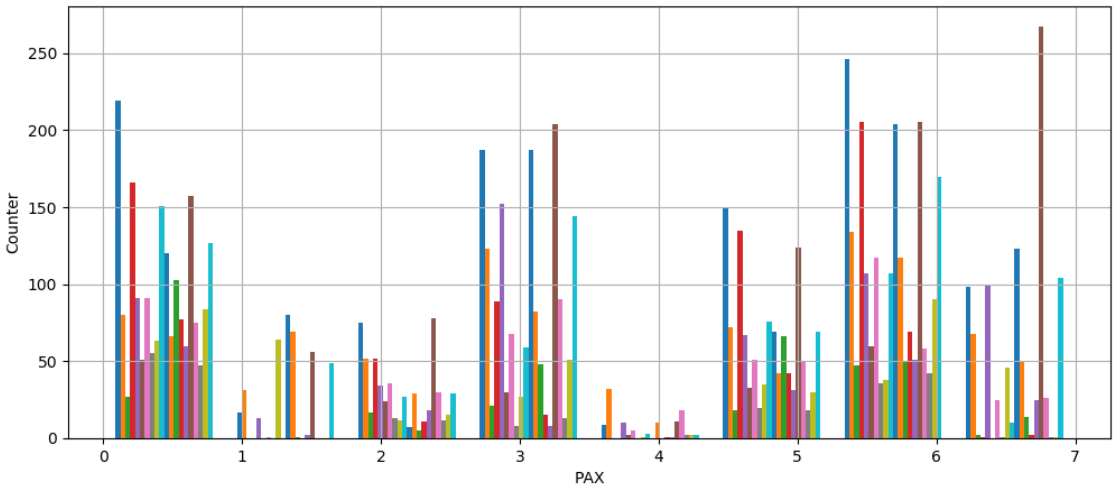

Figure 9: Κατανομή αριθμού πτήσεων ανα σταθμό προορισμού ανα κατηγορία

Ένα ακόμα στοιχείο που βλέπουμε από τα παραπάνω ότι υπάρχει κάποια συσχέτιση μεταξύ κατηγορίας με τα δρομολόγια των πτήσεων. Το γεγονός ότι κάποιες κατηγορίες εκτελούν συγκεκριμένα δρομολόγια τις κάνει πιο ξεχωριστές κατά την ανάλυση.

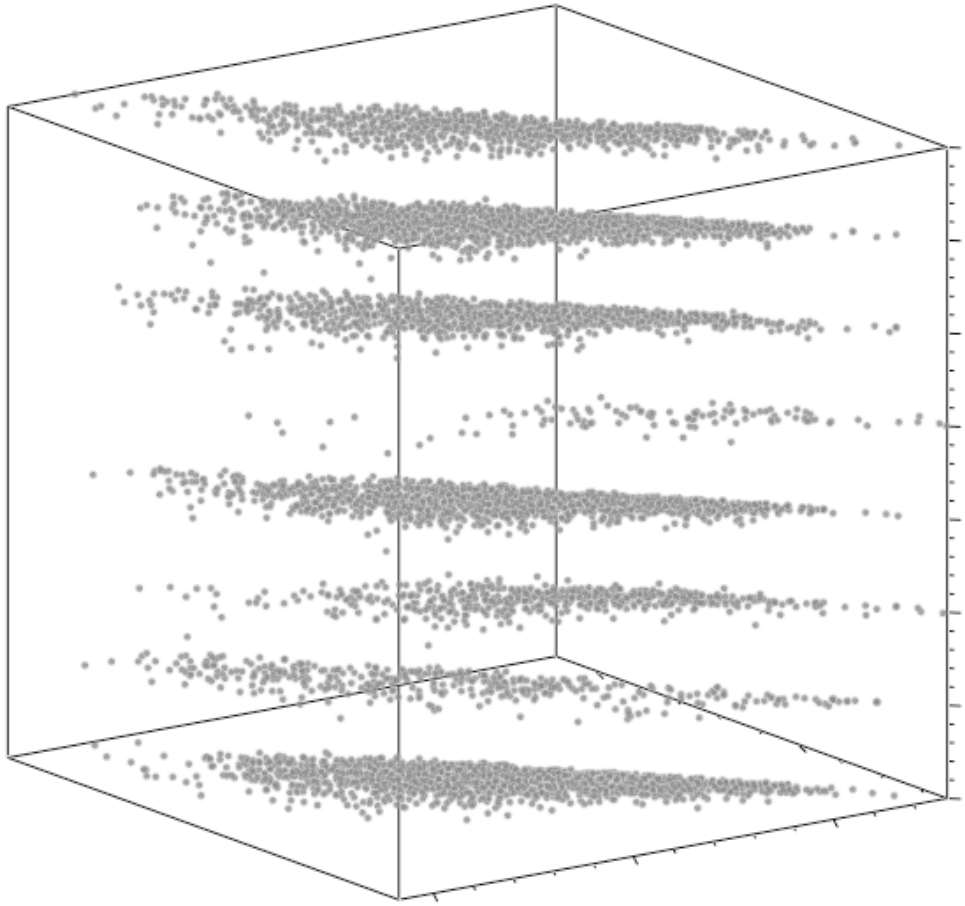

Figure 10: 3d scatterplot WeeksToDeparture, stdwtd, PAX

Από το Figure 10 βλέπουμε ενα μοτίβο τη σχέση μεταξύ WeeksToDeparture και stdwtd ανα κατηγορία πτήσης. Αυτό σημαίνει αρκετές επικαλύψεις αν το προβάλουμε σε δυσδιάστατο χώρο. Επίσης οι αραιές περιεχός ωφείλονται σε λίγα δείγματα για την συγκεκριμένη κατηγορία.

4 Μετασχηματισμοί

Το γεγονός ότι ο υπολογιστής δεν μπορεί να καταλάβει τι σημαίνει ημερομήνια, τοποθεσία κτλ. πρέπει να προσαρμόσουμε τα δεδομένα μας με τέτοιο τρόπο ώστε τα εργαλεία που θα χρησιμοποιήσουμε να είναι αποδότικά και με τον δικό τους τρόπο να ξεχωρίσουν και να κατανοήσουν τι επεξεργάζονται.

4.1 Δημιουργία νέων χαρακτηριστικων

Όπως παρουσιάστηκε στην προηγούμενη ενότητα η ημερομηνία έχει σημαντικό ρόλο στην κατηγορία της πτήσης. Το πρόβλημα είναι οτι στα δεδομένα μας είναι αποθηκευμένη με τρόπο που δεν θα αναγνωριστεί η βαρύτητα από τα μοντέλα μας.

Η τροποποίηση που έγινε, η οποία και χρησιμοποιήθηκε στα γραφήματα, είναι διάσπαση την στήλης DateofDeparture στις επιμέρους FYear, FMonth, FDay (F=Flight). Με αυτο από 1 χαρακτηριστικό δημιουργήσαμε 3 νέα. Επιπλέον δημιουργήθηκε το πεδίο FD06 που απεικονίζει την μέρα της βδομάδας όπου παίρνει τιμές 0-6. Τελευταίο για τις ημερομηνίες είναι η εύρεση την εβδομάδας του χρόνου που πραγματοποιείται η πτήση, τιμες 0-51. Τέλος θα βρούμε, βάση των μηνών σε πιο τετράμηνο του χρόνου πραγματοποιείται η πτήση, τιμές 1-3. Τελικά από την ημερομηνία αναχώρισης έχουμε δημιουργήσει 6 νέες στήλες [FYear,FMonth,FDay,FD06,WeekOfYear,Trimester].
Επόμενο βήμα είναι η ανάλυση των γεωμετρικών συντεγμένων. Η απόσταση που θα διανύσει η πτήση επηρρεάζει αρκετά τον αριθμό επιβατων. Από αυτό θα εξάγουμε την απόσταση του προορισμού και της αναχώρισης βάση απλής ευκλείδειας απόστασης σημείων. Δημιουργείται λοιπόν η στήλη Distance που περιέχει τις αποστάσεις.

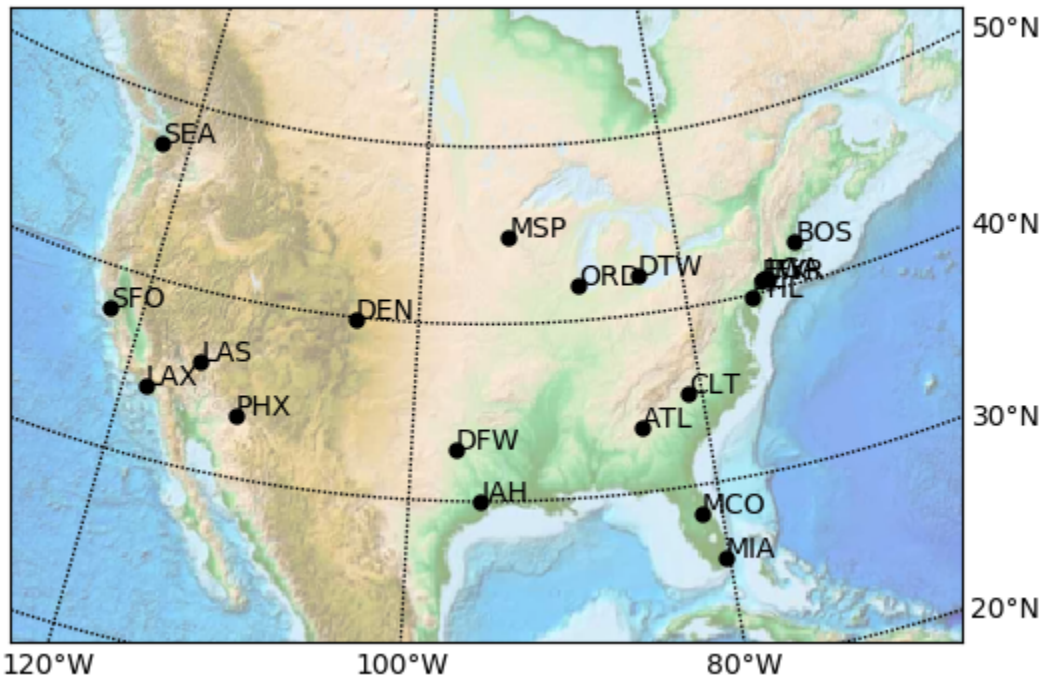

Figure 11: Σημεία στο χάρτη βάση `lat_long` (Basemap)

Βλέποντας το παραπάνω figure, βλέπουμε ότι μπορούμε να χωρίσουμε το χάρτη σε τμήματα. Τα χωρίσματα θα γίνουν για την απλή πέριπτωση και θα βασιστούν στα 40°N, 100°W. Με αυτόν τον τρόπο θα έχουμε σημεία που είναι βορειοδυτικά(NW), βορειοανατολικά(NE), νοτιοδυτικά(SW), νοτιοανατολικά(SE). Αυτό θα το εφαρμόσουμε στο Departure και Arrival. Έτσι δημιουργούμε τις στήλες Start για το Departure, και Finish για το Arrival.

4.2 Επεξεργασία/Προετοιμασία

Επιπλέον θα συμπεριλάβουμε τις στήλες Departure, Arrival, CityDeparture και CityArrival. Λόγω ότι οι παραπάνω στήλες περιέχουν κατηγορικες μεταβλητές, χρειαζόμαστε να τις κωδικοποιήσουμε. Αυτό το κάνουμε με τον LabelEncoder [1] του scikit-learn [2]. Με αυτόν τον τρόπο θα συμβολίσουμε τα στοιχεία ανα στήλη με αριθμούς 0 ως μεγεθος μοναδικών σε αυτήν. Τέλος θα δημιουργήσουμε διανύσματα για κάθε εγγραφή με τη βοήθεια του OneHotEncoder [3].
Ο OneHotEncoder λαμβάνει υπόψην κατηγορικά αριθμητικά χαρακτηριστικά που έχουν παραχθεί είτε από τον LabelEncoder είτε υπάρχοντα και χτίζει ένα διάνυσμα βάση το μέγεθος μοναδικών τιμών που έχει κάθε χαρακτηριστικό.
Για παράδειγμα, έστω ότι κάποια εγγραφή που έχει στη στήλη FMonth την τιμή 5. Το υποδιάνυσμα του τελικού είναι :

$$FMonth = \underbrace{\left[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\right]}_{\text{size }=12}$$

Για να συνοψίσουμε, έχουμε την εξής στήλες για τα δεδομένα μας.

N/N	Feature	Unique
1.	FYear	3
2.	FMonth	12
3.	FDay	31
4.	FD06	7
5.	Departure	20
6.	CityDeparture	19
7.	Arrival	20
8.	CityArrival	19
9.	WeekOfYear	52
10.	Trimester	3
11.	Distance	64
12.	Start	4
13.	Finish	4
X.	Sum	258

Επιπλέον κάθε εγγραφή θα αναπαριστάται ως διάνυσμα της μορφής :

$$Row = \underbrace{\left[\underbrace{000}_{3}\ \underbrace{00...0}_{12}\ \underbrace{00...0}_{31}\ \underbrace{00...0}_{7}\ \underbrace{00...0}_{20}\ \underbrace{00...0}_{19}\ \underbrace{00...0}_{20}\ \underbrace{00...0}_{19}\ \underbrace{00...0}_{52}\ \underbrace{000}_{3}\ \underbrace{00...0}_{64}\ \underbrace{0000}_{4}\ \underbrace{0000}_{4}\right]}_{\text{size=258}}$$

5 Πειράματα

Θα επικεντρωθούμε σε τρεις βασικές μεθόδους που δώθηκε μεγαλύτερο βάρος λόγω απόδοσης. Στην ενότητα 6.2 αναγράφονται μέθοδοι χρησιμοποιήθηκαν αλλά δεν έδωσαν καλύτερο αποτέλεσμα.

5.1 Logistic Regression

Ένας από τους κλασικούς αλγόριθμους για προβλήματα classification. Σε αυτό το στάδιο η διάσταση των δεδομενων ήταν πολύ μικρότερη από την τελική. Αυτό λόγω αντικατάστασης πριν μελετηθεί η δημιουργία νέων χαρακτηριστικών. Υπερπαραμέτροι που χρησιμοποιήθηκαν:

HyperParameter	Description
C	Μεταβλητή κανονικοποίησης
`max_iter`	Μέγιστος αριθμός επαναλήψεων
tol	Συνθήκης τερματισμού σε προσέγγιση
solver	Αλγόριθμος βελτιστοποίσης
`multi_class`	Μέθοδος διαχωρισμού χώρου
`fit_intercept`	Προσθήκη bias

Βάση το LogisticRegression UserGuide [4] η χρήση Newton-cg βοηθάει στη γρηγορότερη σύγκλιση για πολυδιάστατα δεδομένα, σε αυτό το σημείο η διάσταση των δεδομένων ήταν 117 [FYear, FMonth, FDay, DF06, Distance]. Επιπλέον η επιλογή multinomial σαν παράμετρο `mutli_class` ευνοεί τα multiclass προβλήματα.

Παραλαγές στις υπερπαραμέτρους έδινε εύρος αποτελεσμάτων `0.35-0.55` σε f1 score. Ευνοεί μεγάλη διάσταση δεδομένων αλλά επικαλύφθηκε από τις δυνατότητες του SVM.

5.1.1 Αποτελέσματα

Παραθέτονται μερικά αποτελέσματα σε διάφορες περιπτώσεις που συναντήθηκαν κατα τον πειραματισμό. Οι τιμές των παραμέτρων δίνονται με την ακόλουθη σειρά.
1.C, 2.`max_iter`, 3.tol, 4.solver, 5.`multi_class`, 6.`fit_intercept`

1. Περίπτωση label encode DateofDeparture, Departure, Arrival

Values	Dim	Public F1	Private F1
1, 1000, 1e-7, lbfgs, ovr, False	590	0.39970	0.39205
5, 2500, 1e-7, newton-cg, multinomial, False	590	0.43263	0.39654
10, 10000, 1e-7, newton-cg, multinomial, False	590	0.43562	0.39782

2. Περίπτωση label encode FYear, FMonth, FDay, FD06, Departure, Arrival και παραλαγες συνδυασμού FYear, FMonth, FDay, FD06

Values	Dim	Public F1	Private F1
5.0, 1000000, 1e-7, newton-cg, auto, False	133	0.44461	0.41575
20.0, 20000, 1e-6, liblinear, auto, False	87	0.44610	0.41127
5.0, 1000000, 1e-7, newton-cg, auto, True	87	0.44311	0.40871

3. Περίπτωση FMonth, FDay, FD06, Distance

Values	Dim	Public F1	Private F1
17, 100000, 1e-6, newton-cg, multinomial, False	114	0.52544	0.49391

Σε αυτό το στάδιο βλέπουμε ότι η ανάδειξη σημαντικών χαρακτηριστικών ευνοεί την ακρίβεια του αλγορίθμου άσχετα αν το score είναι χαμηλό.

5.2 Support Vector Machines

Μεγάλο άλμα στην ακρίβεια έδωσε η χρήση των Support Vector Machines [5].

HyperParameter	Description
kernel	είδος πυρήνα
gamma	συντελεστής πυρήνα
C	Μέγεθος ποινής κατά λανθάνουσα ταξικοποίησης
tol	Συνθήκη τερματισμού σε προσέγγιση

Συγκεκριμένα χρησιμοποιήθηκε C-Support Vector Classification (SVC). Η μέθοδος που ακολουθούν είναι η One-against-One. Δηλαδή προσπαθούν να διαχωρίσουν μία κλάση απέναντι στις υπόλοιπες μια προς μία. Δηλαδή ένας classifier ανα ξεχωριστή κλάση. Σύνολο κατασκευάζονται N*(N-1)/2 ταξινομητές όπου N αριθμός των κλάσεων.

Επιπλέον η χρήση του πυρήνα παίζει βασικό ρόλο στην ακρίβεια του μοντέλου. Αν τα δεδομένα δεν είναι γραμμικώς διαχωρίσιμα το SVM θα αστοχήσει αρκετά. Μεγάλη απόδοση είναι ο πυρήνας RBF που περικλίει τα δεδομένα σε περιοχές μη γραμμικές, δίνοντας μεγαλύτερη ακρίβεια από ότι ο πολυωνυμικός πυρήνας.

Σχετικά με τις παραμέτρους του πυρήνα [6]. Λόγω της φύσης των δεδομένων, μη ομοιόμορφο μέγεθος δείγματος ανά τάξη, το gamma, που καθορίζει το έυρος επηρροής ενός δείγματος στο χώρο, οποιαδήποτε τιμή εκτός από scaled, έδινε κακό αποτέλεσμα. Το gamma υπολογίζεται ως

$$\gamma = \frac{1}{N_f * \sigma} \quad \text{, where } \sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2},\ N_f = \text{number of features in X, X = train matrix}$$

Αν η μεταβλητή gamma είναι αρκετά ψηλή, η επηρροή του διανύσματος υποστήριξης ακριβώς ο εαυτός του, και η μεταβλητή C δεν θα εμποδίσει το overfitting ότι τιμή και να έχει. Αντίθετα αν είναι πολύ μικρή θα υπάρχει επικάλυψει με τα υπόλοιπα διανύσματα οπότε το σχήμα των δεδομένων δεν θα μπορεί να κατανοηθεί σωστά.

Η μεταβλητή C επηρρεάζει την πολυπλοκότητα της συνάρτησης απόφασης. Καθορίζει την απόσταση των διανυσμάτων υποστήριξων των κλάσεων στον χώρο. Για μικρές τιμές τα διανύσματα μικρότερες αποστάσης, άρα μεγαλύτερη γενίκευση πρόβλεψης αλλά θυσιάζωντας ακρίβεια. Εναλλακτικά για μεγάλες τιμές μεγιστοποιείται η απόσταση των διανυσμάτων οπότε έχουμε πιο αυστηρή πρόβλεψη.

5.2.1 Αποτελέσματα

Οι τιμές των παραμέτρων δίνονται με την ακόλουθη σειρά.
1.C, 2.kernel, 3.gamma, 4.tol

1. Περίπτωση FYear, FMonth, FDay, Distance

Values	Dim	Public F1	Private F1
2.0, notspecified, scale, 1e-5	110	0.47754	0.45099
5.0, rbf, scale, 1e-6	110	0.51347	0.50224
10.0, rbf, scale, 1e-6	110	0.53892	0.52402
21.0, rbf, scale, 1e-6	110	0.55089	0.53235

Στον παρακάτω πινακα έτυχαν μερικες ειδικές περιπτώσεις αφού καταγράφηκε το private score για αυτές. (values constant C=17.0, kernel='rbf', gamma='scale', tol=$1e{-}6$)

Περίπτωση	Dim	Public F1	Private F1
FYear,FMonth,FDay,FD06,Distance	117	0.56586	0.56822
FMonth,FDay,FD06,Distance	114	0.57035	0.56117
FMonth,FD06,Distance	83	0.53592	0.51249
FYear,FMonth,FD06,Distance	86	0.52694	0.52017

Είχε θεωρηθεί ότι το έτος δεν επηρρεάζει σωστά την ταξικοποίηση λόγω ότι έδινε χαμηλότερο σκορ όταν χρησιμοποιώταν στην εκπαίδευση. Όμως με βάση τον πίνακα πετύχαμε καλύτερη γενίκευση και δεν είχαμε μεγάλη απόκλιση προβλέψεων. Αντίθετα ενώ πετυχαίναμε καλύτερο σκορ στο public στην πραγματικότητα κάναμε overffiting λόγω ομοιοτήτων στις περιπτώσεις που το χαρακτηριστικό αυτό έπαιζε ρόλο. Παρόλαυτα το μοντέλο έδειξε αρκετή σταθερότητα και όπως φανεί σε παρακάτω ενότητα, με αρκετή βελτιστοποίηση μπορεί να πετύχει αρκετά υψηλή ακρίβεια.

5.3 Neural Network (Keras-Tensorflow)

Τέλος θα μελετήσουμε την περίπτωση χρήσης νευρωνικών δικτύων. Εδώ έχουμε άφθονες επιλογές για να κατασκευάσουμε το μοντέλο πρόβλεψης. Από τον αριθμό των hiddel layers, αριθμό νευρώνων για κάθε ένα από αυτά, συναρτήσεις βελτιστοποίσης, loss functions, regularizers, συναρτήσεις βελτιστοποίσης και αρκετά άλλα. Καθοριστικό ρόλο βέβαια παίζει η δομή των δεδομένων και το μέγεθος της.

Για λόγους συντομίας θα επιγκεντρωθούμε στα μοντέλα που οι προβλέψεις τους δώθηκαν ως υποβολές, με την τελική μορφή των δεδομένων, ενώ παρακάτω θα καταγραφούν εν συντομία οι αρχικές μορφές τους. Έγινε εκτενής χρήση του Keras[7][8] και του Tensorflow-gpu [9], για να χτίσουμε το μοντέλο μας.

Δυστηχώς όπως φάνηκε και τα δυο μοντέλα υπόκεινταν σε overiffiting. Οι επιλογή υποβολής έγινε αφελώς βάση των υψηλότερων public score. Στην πραγματικότητα υπήρχαν 12 καλύτερες προβλέψεις από οτι αυτή που αναδείχθηκε, γενονός που δείχνει ότι είχαν καλύτερη γενίκευση. Ξανασημειώνεται ότι τα δεδομένα ήταν ανάλογα σε κατανομές χαρακτηριστηκών οπότε ήταν εύλογο να είχαν παρόμοια score.

Λόγω classification χρησιμοποιήθηκε η softmax ως συνάρτηση ενεργοποίησης και η categorical crossentropy σαν loss function. Optimizer ο RMProp με τις default τιμές. Το μοντέλο σε όλες τις περιπτώσεις συμπεριφερόταν ομάλά με batch size = 256. Επιπλέον δινόταν και το διάνυσμα βαρών που υπολογιζόταν από το άθροισμα των εμφανίσεων ανα κατηγορία διαιρεμένο με τον ελάχιστο αυτών για να κανονικοποίηση.

Μοντέλο 1: public score 0,66467, private score 0.61050

Layer	Activation	Size	Dropout
Input	–	258	–
Dense	Relu	224	–
Dropout	–	–	0.7
Dense	Relu	128	–
Dropout	–	–	0.6
Output	softmax	8	–

Βασισμένος στις τοπικές προβλέψεις εσφαλμένα υψηλό ποσοστό Dropout οδήγησε σε αυτό το αποτέλεσμα. Πολλές από τις υποβολές που δώθηκαν βάση τροποποιήσεων είτε ελάχιστη αλλαγή σε μέγεθος κρυφών επιπέδων ή ποσοστού Dropout είχαν παρόμοια αποτελέσματα.

Σαν τελευταίοι πειραματισμού ήταν ο συνδυασμός Dropout και Regularization. Τυχαία επιχειρήθηκε η παρακάτω αρχιτεκτονική.
Μοντέλο 2: public score 0,66017, private score 0.61691

Layer	Activation	Size	Regularizer	Dropout
Input	–	258	–	–
Dense	Relu	224	–	–
Dropout	–	–	–	0.6
Dense	Relu	128	L2(0.001)	–
Dense	Relu	64	L2(0.001)	–
Output	softmax	8	–	–

Ο συνδυασμός των δύο στο ίδιο επίπεδο ήταν καταστροφικός στην πρόβλεψη οπότε παραλείπεται η αναφορά αυτή. Η σκέψη ήταν στο πρώτο επίπεδο να γίνεται ένα φιλτράρισμα των εισερχόμενων σημάτων με τυχαία απενεργοποίση νευρώνων. Όσα σήματα είχαν την δυνατότητα να περάσουν θα κανονικοποιούνταν μέσω των δυο επόμενων επιπέδων ώστε στο τέλος να έχουμε μια ομαλότητα βαρών κατά την εκπαίδευση. Το διάνυσμα είναι αρκετά αραιό και οι άσσοι συνήθως είναι συγκεντρωμένοι. Το θελητό αποτέλεσμα είναι να εχουμε μεγάλο bias ή μικρή πιθανότητα ενεργοποίσης κάποιου νευρώνα που θα πρέπει να ενεργοποιηθεί.
Παρόλαυτα και σε αυτή την περίπτωση το μοντέλο χρειάζεται βελτιστοποίηση διότι υπάρχει μεγάλη απόσταση μεταξύ των score των προβλέψεων ¿0.04.

5.3.1 Ενδεικτικά αποτελέσματα

Μερικά αποτέλεσματα από μοντέλα που δεν επιλέχθηκαν
Model 1.
Features: FYear FMonth FDay FD06 Departure Arrival Distance = 157
Model: 157(Input)-¿[8(Relu)-¿8(Relu)]-¿8(Softmax)
optimizer: Default Adam, loss: categorical crossentropy
fit: epochs=200, batch size = 5
Neural scores: Public: 0.54940 Private: 0.54067
Model 2.
Features: FYear FMonth FDay FD06 Departure Arrival Distance = 157
Model: 157(Input)-¿[157(Relu)-¿Dropout(0.5)-¿8(Relu)-¿Dropout(0.5)]-¿8(Softmax)
optimizer: Default SGD, loss: categorical crossentropy
fit: epochs=130, batch size = 256
Neural scores: Public: 0.62125 Private: 0.60409
Model 3.
Features: FYear FMonth FDay FD06 Departure Arrival Distance = 157
Model: 157(Input)-¿[157(Relu)-¿Dropout(0.5)-¿8(Relu)-¿Dropout(0.5)]-¿8(Softmax)
optimizer: Default SGD, loss: categorical crossentropy
fit: epochs=130, batch size = 256
Neural scores: Public: 0.62125 Private: 0.60409
Model 4.
Features: FYear FMonth FDay FD06 Dep CDep Arr CArr Distance Start Finish = 203
Model: 157(Input)-¿[157(Relu)-¿Dropout(0.6)-¿32(Relu)-¿Dropout(0.5)]-¿8(Softmax)
optimizer: Default SGD, loss: categorical crossentropy
fit: epochs=130, batch size = 256
Neural scores: Public: 0.61676 Private: 0.58360

6 Αποσυρθέντα

Παρακάτω καταγράφονται συνοπτικά μετασχηματισμού και μέθοδοι που δεν είχαν θετική επίδραση στα τοπικά και τελικά αποτελέσματα ή που τελικά αντικαταστάθηκαν από αυτά που περιγράφονται στην ενότητα 5 λόγω κακής επίδοσης.

6.1 Μετασχηματισμοί

1. Concantination στηλών Αναχώρισης και Άφιξης.
2. Concantination στηλών LatLong Αναχώρισης.
3. Concantination στηλών LatLong Άφιξης.
4. Ενσωμάτωση ακεραιου μέρους μέσου όρου δέσμευσης πτήσης.
5. Ενσωμάτωση ακέραιου μέρους τυπικής απόκλισης μέσου όρου.
6. Υπολογισμός διαφοράς ημερομηνίας δεσμευσης πτήσης από ημερομηνία πραγματοποίσης πτήση μέ χρήση μέσου όρου.
7. Επιλογή χρήσης επιμέρους τμήματος των στηλών FYear, FMonth, FDay.
8. Δημιουργίας στήλης διαστήματος μέσου όρου βάση διαστημάτων εμπιστοσύνης 0.5%, 0.6%, 0.7%, 0.8%, 0.9%, 0.95% ανα κατηγορία πτήσης.
9. Διαγραφή εγγραφών εκτός των διαστήματος εμπιστοσύνης
10. Δημιουργία σύνθετου χαρακτηριστικού βάση γεωγραφικών συντεταγμένων με σύμβολα N, S, W, E, C για αναχώριση και άφιξη.
11. Εύρεση μέσης ημέρας κράτησης βάση μέσο όρο και ημέρα πτήσης
12. Υπολογισμός πλήρης ημερομηνίας κράτησης βάση μέρο όρο
13. Εύρεση απόστασης εβδομάδων από ημερομηνία πτήσης βάση μέσο όρο
14. Αποκείνηση ημερών ως ακριβείς ημέρες έτους (1-365)
15. Χώρισμα έτους σε εξάμηνα (2.)
16. Χώρισμα έτους σε τρίμηνα (4.)

17+. Πολυάριθμοι συνδυασμοί δοθέντων ή και παραγώμενων στηλών (χαρακτηριστικών) με τελικό αριθμό χαρακτηριστικών ¿300.

6.2 Μέθοδοι

*Οι παρακάτω μέθοδοι δεν είναι βελτιστοποιημένοι, ούτε είχαν βαση την τελική μορφή των δεδομένων, το οποίο είναι διπλάσιο σε χαρακτηριστικά από ότι είχαν δοκιμαστεί. Το αποτέλεσμα του F1 είναι το τελευταίο που εμφανίστηκε πριν αντικατασταθούν.

Model	Public F1	Private F1
SGDClassifier	0.45059	N/A
RidgeClassifier	0.36976	0.37475
LinearSVC	0.39970	0.39718
DecisionTreeClassfier	0.50149	N/A
RandomForestClassifier	0.51796	0.50992
ExtraTreeClassifier	0.44760	N/A

7 Συμπεράσματα

Προσωπικό σημείο εστίασης είναι η σύγκριση επίδοσης μεταξύ SVM και NeuralNetworks. Είχαν αρκετά παραπλήσια επίδοση με το SVM να έρχεται 2ο, τουλάχιστον με την βέλτιστη του μορφή τη δεδομένη στιγμή. Αλλά η πολυπλοκότητα υλοποίσης του είναι αρκετά μικρότερη και οι επιλογές για

τις υπερπαραμέτρους περιορισμένες σε σύγκριση με την τυχαιότητα που έχει κάποιο νευρωνικό δίκτυο. Αυτό συνέβαινε τουλάχιστον στη τωρινή περίπτωση.

Γενικότερα όμως στα προβλήματα ταξικοποιησης που μελετήσαμε αξίζει χρησιμοποιούνται απλές μέθοδοι σχεδίασης μοντέλου πριν προβούμαι σε πολύπλοκες και πλήρης ελευθέρας βούλησης σχεδιασμού.
Παραθέτονται μερικά συγκρητικά αποτέλέσματα μεταξύ SVM και neuralnet.
SVM parameters constant : kernel='rbf', gamma='scale', C=17

Περίπτωση 1.
Features: FYear FMonth FDay FD06 Departure Arrival Distance = 157
Model: 157(Input)-¿[157(Relu)-¿Dropout(0.6)-¿16(Relu)-¿Drop(0.5)]-¿8(Softmax)
optimizer: Default SGD, loss: categorical crossentropy
fit: epochs=200, batch size = 128
Neural scores: Public: 0.61077 Private: 0.59000
SVM scores: Public: 0.60479 Private: 0.57783

Περίπτωση 2.
Παρόμοιο με την περίπτωση 1 αλλά παράλειψη FYear
Features: FMonth FDay FD06 Departure Arrival Distance = 154
Neural scores: Public: 0.57035 Private: 0.56438
SVM scores: Public: 0.57185 Private :0.55541

Περίπτωση 3.
Features: FYear FMonth FDay FD06 Departure Arrival Distance = 157
Model: 157(Input)-¿[157(Relu)-¿Dropout(0.5)-¿32(Relu)-¿Drop(0.6)]-¿8(Softmax)
optimizer: Default SGD, loss: categorical crossentropy
fit: epochs=300, batch size = 256
Neural scores: Public: 0.62275 Private: 0.60345
SVM scores: Public: 0.57784 Private: 0.56758

Κάτι που δεν υπήρχει χρόνος να μελετηθεί είναι ο έλεγχος υπολογισμού βαρύτητας νέων χαρακτηριστικών μέσω regression όπως και το αν το μέγεθος του sparsity των τιμών 1 σε κάθε διάνυσμα του δείγματος επηρρεάζει τα μοντέλα στο να αναγνωρίζει καλύτερα τα δεδομένα.

8 References

[1] Label Encoder
`https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html`.

[2] Scikit Learn.
`https://scikit-learn.org/stable/index.html`.

[3] One Hot Encoder.
`https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html`.

[4] Logistic Regression.
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.

[5] Support Vector Machines
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[6] RBF SVM parameters
https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

[7] Keras Chollet, François and others, 2015 GitHub: https://github.com/rstudio/keras

[8] R Interface to Keras. Chollet, François and Allaire, JJ and others, 2017 https://keras.io

[9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.