Πανεπιστήμιο Μακεδονίας
Τμήμα Εφαρμοσμένης Πληροφορικής

Τίτλος Εργασίας:

«1η Εργασία – Classification Problems»

Συντάκτες:

Ευθυμίου Βασίλειος[1], **Παπαδόπουλος Νικόλαος**[2], **Τζελαλής Γεώργιος**[3]
& Τσώνη Σταυρούλα[4]

Διδάσκων: Πρωτοπαπαδάκης Ευτύχιος
Μάθημα: Μηχανική Μάθηση
Εξάμηνο Μαθήματος: 7ο
Ημερομηνία: 22/12/2023

[1]**ics21035**@uom.edu.gr
[2]**ics21048**@uom.edu.gr
[3]**ics21032**@uom.edu.gr
[4]**ics21074**@uom.edu.gr

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή

Η παρούσα έρευνα επικεντρώνεται στο πρόβλημα του εντοπισμού εταιρειών που πρόκειται να κηρύξουν χρεοκοπία, χρησιμοποιώντας έτοιμα δεδομένα που παρέχονται από αρμόδιο οργανισμό. Στο πλαίσιο αυτό, σκοπός της έρευνας είναι η παρουσίαση λεπτομερών συγκριτικών αποτελεσμάτων, αναφορικά με την ικανότητα διαφόρων τεχνικών ταξινόμησης να ανταπεξέλθουν στο συγκεκριμένο πρόβλημα.

Στο πλαίσιο της εργασίας, εφαρμόσαμε μοντέλα ταξινόμησης, τα οποία εξετάζουμε λεπτομερώς στις επόμενες ενότητες. Η ανάλυση περιλαμβάνει τόσο τη θεωρητική βάση των μεθόδων που χρησιμοποιήσαμε, όσο και τα πειραματικά αποτελέσματα που προέκυψαν από την εφαρμογή των μοντέλων μας στα παρασχεθέντα δεδομένα.

Έπεται μια σύντομη περιγραφή των κεντρικών σημείων της αναφοράς που θα αναπτυχθούν στις επόμενες ενότητες. Η αναφορά ακολουθεί μια ολοκληρωμένη δομή, προσφέροντας στον αναγνώστη μια συνολική εικόνα της εργασίας μας. Στην επόμενη ενότητα, «*Θεωρητικό Υπόβαθρο*», αναλύουμε λεπτομερώς τις μεθόδους που εφαρμόστηκαν, εστιάζοντας στις παραμέτρους που επηρεάζουν την απόδοση των μοντέλων. Στη συνέχεια, η ενότητα «*Πειραματικά Αποτελέσματα*» προσφέρει μία εκτενή επισκόπηση των συλλεγέντων δεδομένων, παρουσιάζοντας τα αποτελέσματα με τη μορφή πινάκων, γραφικών παραστάσεων και εικόνων. Τέλος, η ενότητα «*Συμπεράσματα*» επισημαίνει τα κύρια ευρήματα της έρευνας. Εδώ, συγκρίνουμε την απόδοση των διαφόρων μοντέλων και προτείνουμε το πιο αποτελεσματικό.

2. Θεωρητικό Υπόβαθρο

Σε αυτό το τμήμα της αναφοράς θα αναλύσουμε τις μεθόδους που χρησιμοποιήθηκαν για την αντιμετώπιση του παραπάνω ζητήματος. Ζητούμενο των μοντέλων ταξινόμησης είναι να κατηγοριοποιήσουν μια εταιρεία ως υγιή ή χρεοκοπημένη. Η σωστή κατηγοριοποίηση μιας χρεοκοπημένης εταιρίας είναι σημαντικότερη από τη σωστή κατηγοριοποίηση μιας υγιούς, διότι αν μια εταιρία που πρόκειται να χρεοκοπήσει καταταχθεί ως υγιής θα συνεχίσει να λειτουργεί ως πρότινος, αυξάνοντας ενδεχομένως κι άλλο τα χρέη της και δυσχεραίνοντας την κατάστασή της. Αντιθέτως, αν μια υγιής επιχείρηση κατηγοριοποιηθεί ως χρεοκοπημένη μπορεί να οδηγηθεί απλά σε περιττό οικονομικό έλεγχο. Επομένως η μετρική που χρήζει ιδιαίτερης προσοχής είναι το recall, καθώς δείχνει το ποσοστό των χρεοκοπημένων εταιρειών που προβλέφθηκαν σωστά. Στα πλαίσια αυτής της ανάλυσης θα αναφερθούμε στη λειτουργικότητα των μοντέλων και τις υπερ-παραμέτρους τους, οι οποίες προέκυψαν βάσει της απόδοσης των μοντέλων πάνω στα balanced test δεδομένα, καθώς με unbalanced δεδομένα τα μοντέλα που εκπαιδεύονται έχουν χαμηλότερη απόδοση.

1. Linear Discriminant Analysis

Για την εφαρμογή του μοντέλου Linear Discriminant Analysis (LDA) αξιοποιήθηκε ο συνονόματος classifier που παρέχει η βιβλιοθήκη scikit-learn. Η συμπεριφορά του μοντέλου LDA κατά τη διάρκεια της εκπαίδευσής του μπορεί να επηρεαστεί από τις ακόλουθες υπερ-παραμέτρους: η υπερ-παράμετρος solver που πήρε την τιμή Least squares solution χρησιμεύει στην εκτίμηση των παραμέτρων και επηρεάζει το accuracy του μοντέλου. Η υπερ-παράμετρος shrinkage που σχετίζεται με την κανονικοποίηση των δεδομένων τέθηκε ίση με None, όπως και η n_components η οποία αφορά τη μείωση των διαστάσεων. Επιπλέον, υπάρχει δυνατότητα χειρισμού class imbalance από την υπερ-παράμετρο priors, η οποία όπως και στη μέθοδο *Naive Bayes* τέθηκε ίση με None. Η εκτίμηση των πινάκων covariance έγινε από τον empirical covariance estimator με την υπερ-παράμετρο covariance_estimator και τέλος, η υπερ-παράμετρος tol που αντιμετωπίζει τη σύγκλιση πήρε την τιμή 0.0001.

2. Logistic Regression

Το Logistic Regression (LR) είναι μία μέθοδος μηχανικής μάθησης η οποία υλοποιήθηκε από τη βιβλιοθήκη scikit-learn της Python. Οι υπερ-παράμετροι της LR διαδραματίζουν κρίσιμο ρόλο στην απόδοση του μοντέλου. Ο αλγόριθμος βελτιστοποίησης τέθηκε ως ο liblinear από την παράμετρο solver με τύπο κανονικοποίησης L2 από την Penalty που πήρε την τιμή l2. Η υπερ-παράμετρος tol είναι ίδια με εκείνη του LDA και είναι πάλι ίση με 0.0001. Η random_state τέθηκε ίση με None και εξασφαλίζει την αναπαραγωγιμότητα, ενώ το max_iter που περιορίζει τον αριθμό των επαναλήψεων για τη σύγκλιση, ιδιαίτερα σημαντικό για μεγάλα σύνολα δεδομένων, όπως το συγκεκριμένο,

ορίστηκε ίσο με 200.

3. Decision Tree

Για τη δημιουργία του Decision Tree μοντέλου αξιοποιήθηκε η μέθοδος DecisionTreeClassifier() της βιβλιοθήκης scikit-learn. Χρησιμοποιήθηκαν οι default παράμετροι, πέραν της υπερ-παραμέτρου max_depth, η οποία τέθηκε ίση με 5, διότι έπειτα από δοκιμές παρατηρήθηκε ότι κατά μέσο όρο βελτιώνονται οι τιμές των μετρικών και αποτρέπεται το overfitting πάνω στα training data. Συγκεκριμένα, το μοντέλο με τις default υπερ-παραμέτρους έδινε τις εξής τιμές μετρικών: accuracy = 0.82, precision = 0.51, recall = 0.68, f1 score = 0.47, roc-auc = 0.68, ενώ με max_depth = 5 οι μετρικές έλαβαν τις ακόλουθες τιμές: accuracy = 0.85, precision = 0.51, recall = 0.68, f1 = 0.48, roc-auc = 0.76. Συνεπώς, το accuracy αυξήθηκε κατά 3.66%, τα precision και recall παρέμειναν σταθερά, το f1 score βελτιώθηκε κατά 2.13% και το roc-auc κατά 11.76%.

4. Random Forest

Η δημιουργία του μοντέλου Random Forest έγινε με χρήση της μεθόδου RandomForestClassifier() της βιβλιοθήκης scikit-learn. Έπειτα από δοκιμές οι υπερ-παράμετροι min_samples_split και max_depth τέθηκαν ίσες με 4 και 8 αντίστοιχα, ενώ οι υπόλοιπες διατήρησαν τις default τιμές τους. Η προσθήκη των συγκεκριμένων τιμών στις παραμέτρους αυξάνει κυρίως το recall και αποτρέπει το overfitting πάνω στα training δεδομένα. Οι τιμές των μετρικών πριν την μεταβολή των προαναφερθέντων υπερ-παραμέτρων ήταν οι ακόλουθες: accuracy = 0.925, precision = 518, recall = 0.68, f1 = 52, roc-auc = 0.83, ενώ έπειτα μεταβλήθηκαν ως εξής: accuracy = 0.928, precision = 0.520, recall = 0.70, f1 = 0.52, roc-auc = 0.84. Συνεπώς, το accuracy αυξήθηκε κατά 0.32%, το precision κατά 0.39%, το recall κατά 2.94%, το f1 score παρέμεινε σταθερό και το roc-auc βελτιώθηκε κατά 1.2%.

5. k-Nearest Neighbors

Για το k-Nearest Neighbors (KNN) χρησιμοποιήθηκε η υλοποίηση του KNN που παρέχεται από τη βιβλιοθήκη scikit-learn. Τα hyperparameters που αξιοποιήθηκαν είναι το k, που συμβολίζει τον αριθμό πλησιέστερων γειτόνων. Όπως φαίνεται και στον παραδοτέο κώδικα, για τον αλγόριθμο k-Nearest Neighbors, για την επιλογή του k προσεγγίζουμε μία ισορροπία μεταξύ ακρίβειας και γενίκευσης. Μετά από δοκιμές διαφόρων τιμών στο k, παρατηρήθηκε ότι για k=3 η απόδοση του μοντέλου ήταν η καλύτερη. Μικρότερες τιμές του k οδηγούσαν το μοντέλο σε overfitting, δηλαδή σε μη ικανοποιητική απόδοση λόγω υπερβολικά τοπικής προσαρμογής και αδυναμίας να αναγνωρίσει τη γενικότερη δομή των δεδομένων. Αντιθέτως, μεγαλύτερες τιμές του k οδηγούσαν σε underfitting, όπου το μοντέλο γινόταν πολύ γενικό και έχανε την ικανότητά του να προβλέπει σωστά νέα δεδομένα λόγω της μειωμένης ευαισθησίας του

στις τοπικές ιδιαιτερότητες.

6. Naive Bayes

Για το Naive Bayes (NB) χρησιμοποιήθηκε η υλοποίηση Gaussian Naive Bayes που παρέχεται από τη βιβλιοθήκη scikit-learn. Όπως φαίνεται και στον παραδοτέο κώδικα, δεν έχουν οριστεί συγκεκριμένες προκαταρκτικές πιθανότητες (priors) για τις κλάσεις, οπότε ο αλγόριθμος υπολογίζει αυτές τις πιθανότητες αυτόματα από τα δεδομένα εκπαίδευσης. Αυτή είναι μια κοινή πρακτική, ιδιαίτερα όταν δεν υπάρχει ειδική προηγούμενη γνώση για την κατανομή των κλάσεων.

7. Support Vector Machine

Για το Support Vector Machine (SVM) χρησιμοποιήθηκε η υλοποίηση του SVC (Support Vector Classifier) που παρέχεται από τη βιβλιοθήκη scikit-learn. Τα hyperparameters που αξιοποιήθηκαν είναι τα random_state, C, kernel και probability. Το random_state εξασφαλίζει την αναπαραγωγιμότητα του μοντέλου σε κάθε εκτέλεση, ενώ το default kernel ‘rbf’ φαίνεται να αποδίδει καλύτερα στο συγκεκριμένο dataset σε σύγκριση με τα υπόλοιπα kernels. Μετά από δοκιμές διαφόρων τιμών στο C, φάνηκε πως για C=10 η απόδοση του μοντέλου κορυφώνεται, ενώ για μικρότερες τιμές τείνει να κάνει underfitting και για μεγαλύτερες overfitting κατά τη διάρκεια του training. Τέλος, θέσαμε το probability=True για λόγους συμβατότητας με τις συναρτήσεις που χρησιμοποιούμε παρακάτω για υπολογισμό των μετρικών και σχεδιασμό των confusion matrices.

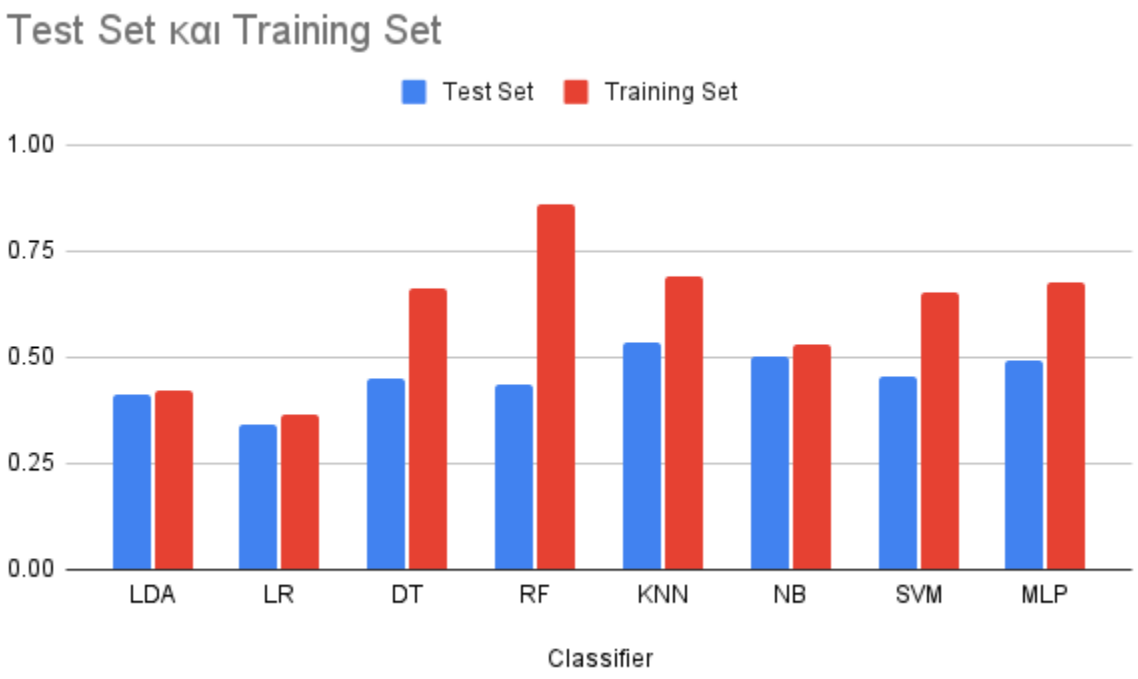
8. Multilayer Perceptron

Στο Multilayer Perceptron (MLP) το random_state έχει την ίδια χρήση με πριν. Για τα hidden layers δοκιμάστηκαν διάφορα μεγέθη και τελικά η χρήση ενός μόνο κρυφού επιπέδου με 12 neurons φαίνεται να επαρκεί στη συγκεκριμένη περίπτωση, χωρίς να υπάρχει ανάγκη για πιο περίπλοκη και σύνθετη αρχιτεκτονική. Για το activation function και το alpha χρησιμοποιήθηκαν οι default τιμές, και ως solver ο “lbfgs” που αποδίδει καλύτερα από τους “sgd” και “adam”, λόγω του μικρού μεγέθους του dataset και του imbalance που αυτό παρουσιάζει. Το max_iter ορίστηκε ως 10.000 για να δώσει περιθώρια στο μοντέλο να συγκλίνει, καθώς μικρότερες τιμές που δοκιμάστηκαν δεν ήταν πάντα επαρκείς. Τέλος, θέσαμε το early_stopping=True για να αποφευχθούν περιπτώσεις overfitting πάνω στα δεδομένα.

3. Πειραματικά Αποτελέσματα

Στο συγκεκριμένο τμήμα της εργασίας θα αναλύσουμε τη συμπεριφορά των παραπάνω μοντέλων που εκπαιδεύτηκαν στο balanced training set με αναλογία μία χρεοκοπημένη προς τρεις υγιείς επιχειρήσεις. Συγκεκριμένα, θα χρησιμοποιήσουμε ως μέτρο σύγκρισης τις μετρικές recall, specificity και F1 score. Τα δεδομένα που προέκυψαν από την εκτέλεση του κώδικα και οι μετρικές που υπολογίστηκαν (και αξιοποιήθηκαν για την κατασκευή των διαγραμμάτων που ακολουθούν) είναι διαθέσιμα σε αυτό το Google Spreadsheet.

1. Recall

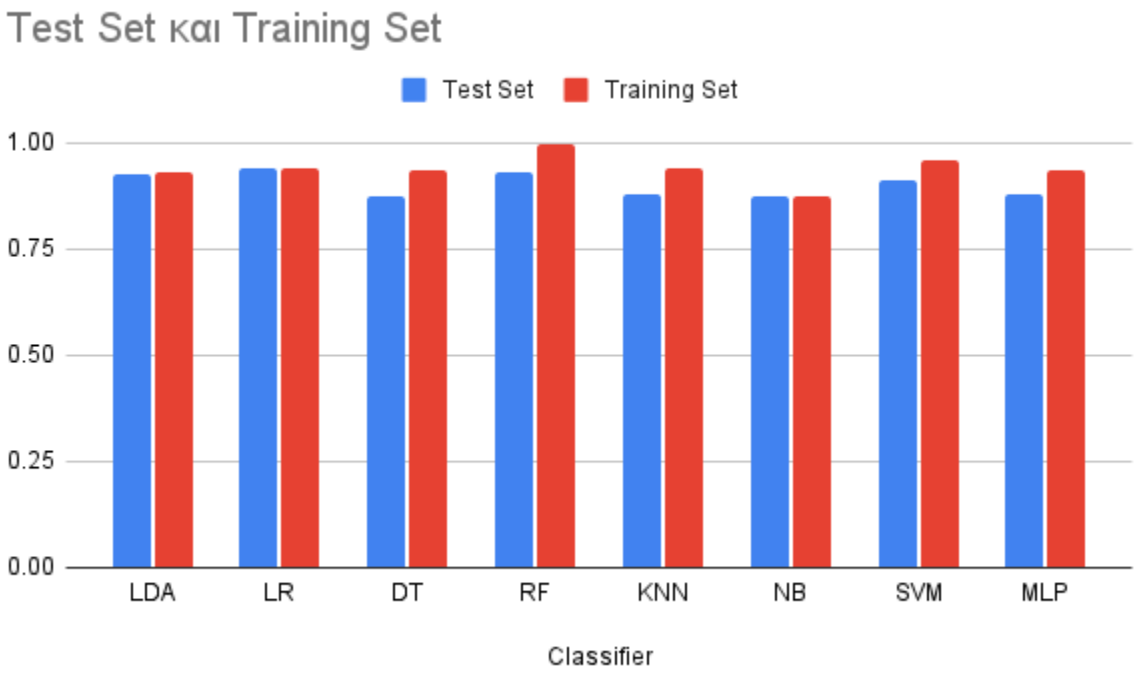


Εικόνα 3.1.1: Recall scores of each Classifier on the Test and Training sets

Το διάγραμμα (*Εικόνα 3.1.1*) απεικονίζει τη μετρική recall (κατά μέσο όρο) για διάφορους classifiers, όπως αυτοί εκπαιδεύτηκαν και αξιολογήθηκαν πάνω στα training set και test set, αντίστοιχα. Το recall είναι ένας δείκτης που απεικονίζει κάτα πόσο το μοντέλο είναι σε θέση να εντοπίσει όλες τις χρεοκοπημένες επιχειρήσεις.
Από το διάγραμμα φαίνεται ότι ο Random Forest (RF) classifier έχει το υψηλότερο recall στο training set. Ωστόσο, στο test set η απόδοση μειώνεται σημαντικά. Ο k-Nearest Neighbors (KNN) classifier έχει τη δεύτερη υψηλότερη τιμή recall στο training set, ενώ την υψηλότερη τιμή στο test set. Οι Linear Discriminant Analysis (LDA), Logistic Regression (LR) και Naive Bayes (NB) classifiers εμφανίζουν σχετικά σταθερή απόδοση μεταξύ των test και training set, με τον Naive Bayes (NB) να έχει την δεύτερη καλύτερη

απόδοση στο test set. Οι Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (KNN), Support Vector Machine (SVM) και Multilayer Perceptron (MLP) classifiers εμφανίζουν υψηλή μείωση του recall καθώς μεταβαίνουμε από το training set στο test set, διατηρώντας ωστόσο μια καλή επίδοση. Το γεγονός αυτό θα μπορούσε να υποδηλώνει overfitting κατά την εκπαίδευση, καθώς δε γενικεύουν τόσο καλά σε νέα δεδομένα.

2. Specificity



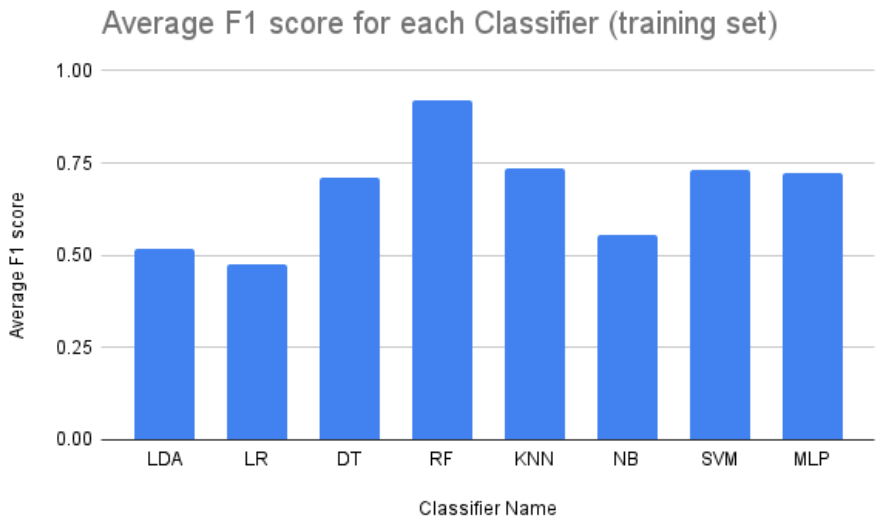
Εικόνα 3.2.1 Specificity scores of each Classifier on the Test and Training sets

Το διάγραμμα (*Εικόνα 3.2.1*) απεικονίζει τη μετρική specificity για διάφορα μοντέλα ταξινόμησης, συγκρίνοντας τις τιμές ανάμεσα στο test set και στο training set. Η μετρική specificity μετρά το ποσοστό των αρνητικών περιπτώσεων που έχουν προβλεφθεί σωστά ως αρνητικές από το μοντέλο (όπου αρνητικές στην περίπτωσή μας θεωρούνται οι υγιείς εταιρείες). Έτσι, ένα υψηλό σκορ specificity υποδηλώνει ότι λίγες αρνητικές περιπτώσεις έχουν καταχωρηθεί λανθασμένα ως θετικές.

Από το διάγραμμα παρατηρούμε ότι, όλα τα μοντέλα παρουσιάζουν υψηλό specificity τόσο στο training set όσο και στο test set, υποδηλώνοντας γενικά καλή ικανότητα στον εντοπισμό των πραγματικά αρνητικών περιπτώσεων. Το μοντέλο Random Forest (RF) φαίνεται να έχει την καλύτερη επίδοση στο training set, ενώ εμφανίζει τη δεύτερη καλύτερη επίδοση στο test set. Ωστόσο, το μοντέλο Logistic Regression (LR) είναι αυτό που υπερισχύει των υπολοίπων στο test set. Οι ταξινομητές Linear Discriminant Analysis (LDA), Logistic Regression (LR) και Naive Bayes (NB) διατηρούν σταθερή επίδοση

μεταξύ των δύο σετ, χωρίς ιδιαίτερες διακυμάνσεις. Παράλληλα, τα μοντέλα Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (KNN), Support Vector Machine (SVM) και Multilayer Perceptron (MLP) δείχνουν ελαφρώς χαμηλότερο specificity στο test set σε σύγκριση με το training set. Η μείωση παραμένει εντός λογικών πλαισίων, οπότε για την αρνητική κλάση των υγιών εταιρειών δεν εμφανίζονται περιστατικά overfitting, όπως συνέβη με τη θετική κλάση.
Στο dataset περιλαμβάνεται ικανοποιητικό πλήθος δειγμάτων υγιών εταιρειών, τα οποία επαρκούν για να μπορέσουν τα μοντέλα να εκπαιδευτούν αποτελεσματικά στον σωστό εντοπισμό τους και να αποφευχθεί overfitting.

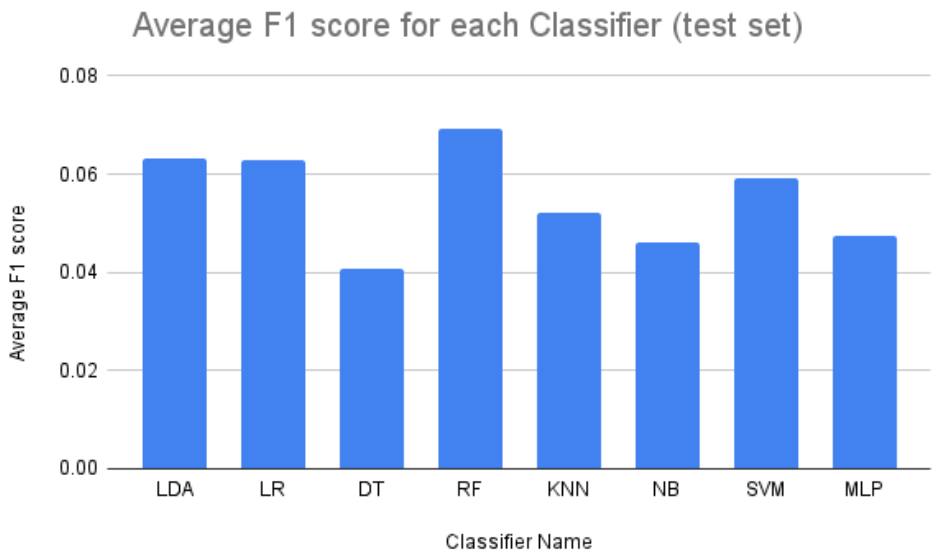
3. F1 Score



Εικόνα 3.3.1: Average F1 score of each classifier (training set)

Στο παραπάνω διάγραμμα, παρουσιάζεται η τιμή του μέσου F1 score για κάθε classifier πάνω στο training set. Το F1 score είναι ένα μέτρο απόδοσης που συνδυάζει τόσο το precision όσο και το recall, προσφέροντας μια ισορροπημένη αξιολόγηση της απόδοσης του εκάστοτε classifier. Από το διάγραμμα φαίνεται ότι το μεγαλύτερο μέσο F1 score με διαφορά το κατέχει ο Random Forest (RF) classifier (0.92). Έπειτα ακολουθούν οι Decision Tree (DT), k-Nearest Neighbor (KNN), Support Vector Machine(SVM) και Multilayer Perceptron (MLP) classifiers οι οποίοι εμφανίζουν παραπλήσια F1 score. Τέλος, τα χαμηλότερα score κατά σειρά έχουν οι Logistic Regression (LR), Linear Discriminant Analysis (LDA) και Naive Bayes (NB). Ωστόσο, είναι σημαντικό να σημειωθεί ότι παρόλο που ορισμένοι classifiers εμφανίζουν υψηλές επιδόσεις στο training set, αντίστοιχη απόδοση στο test set δεν είναι πάντοτε σίγουρη, πράγμα που φαίνεται και στο παρακάτω διάγραμμα (*Εικόνα 3.3.2*). Επιπλέον, ένας classifier με υψηλό F1 score στο training set μπορεί να είναι overfitted στα δεδομένα εκπαίδευσης και

να μην γενικεύει καλά σε νέα δεδομένα. Η αξιολόγηση στο test set θα δώσει πιο συνεπή συμπεράσματα για την πραγματική τους απόδοση.



Εικόνα 3.3.2: Average F1 score of each classifier (test set)

Στο παραπάνω διάγραμμα (*Εικόνα 3.3.2*), παρουσιάζεται η μέση τιμή της μετρικής F1 score για κάθε classifier πάνω στο test set, ενώ στην *Εικόνα 3.1.1* παρουσιάζεται η αντίστοιχη πληροφορία αναφορικά με το training set. Αξίζει να σημειωθεί, πως τα σκορ είναι σημαντικά χαμηλότερα στο test set σε σχέση με το training set, πράγμα που σε ορισμένες περιπτώσεις οφείλεται σε overfitting των μοντέλων (με το φαινόμενο να είναι σε άλλα λιγότερο και σε άλλα περισσότερο έντονο), καθώς και στο αρκετά μικρό μέγεθος του training set, και κυρίως στο μικρό πλήθος διαθέσιμων δειγμάτων για χρεοκοπημένες εταιρείες, που οδηγεί σε δυσκολία των μοντέλων να γενικεύουν σε νέα δεδομένα. Η αδυναμία αυτή εντείνεται ακόμη περισσότερο από το υπερβολικά έντονο imbalance που υπάρχει μεταξύ των δύο κλάσεων στο test set.

Συγκρίνοντας τα μοντέλα, ο Random Forest (RF) classifier παρουσιάζει το υψηλότερο F1 score στο test set, ακολουθούμενος από τους Linear Discriminant Analysis (LDA) και Logistic Regression (LR) και αμέσως μετά τον Support Vector Machine (SVM). Ο Decision Tree (DT) classifier φαίνεται να έχει τη χαμηλότερη απόδοση όλων με βάση το συγκεκριμένο μέτρο σύγκρισης. Παρόμοια, ούτε οι Naive Bayes (NB), k-Nearest Neighbor (KNN) και Multilayer Perceptron (MLP) classifiers παρουσιάζουν ιδιαίτερα υψηλή απόδοση συγκριτικά με τους καλύτερους 4 classifiers που αναφέρθηκαν στην αρχή.

4. Συμπεράσματα

1. Βέλτιστο Μοντέλο

Για την επιλογή του βέλτιστου μοντέλου μεταξύ όσων υλοποιήθηκαν για το παρόν ζητούμενο θα χρησιμοποιήσουμε ως γνώμονα τη μετρική F1 score. Όπως, έχουμε αναφέρει ήδη στο τμήμα «*Πειραματικά Αποτελέσματα*» της αναφοράς, η συγκεκριμένη μετρική προσφέρει μία ισορροπημένη αξιολόγηση της απόδοση των μοντέλων. Για την εύρεση του βέλτιστου μοντέλου θα αναφερθούμε αποκλειστικά στα αποτελέσματα των μοντέλων στο test set, όπως είδαμε και στην *Εικόνα 3.3.2*. Συνοψίζοντας, το βέλτιστο από τα μοντέλα που παρουσιάστηκαν στο τμήμα «*Θεωρητικό Υπόβαθρο*» για το συγκεκριμένο σύνολο δεδομένων είναι το Random Forest. Το Random Forest αποδίδει σταθερά καλύτερα από όλα τα υπόλοιπα μοντέλα, έχοντας την υψηλότερη τιμή στη μετρική F1 score, ίση με 0.07. Επιπρόσθετα, είναι αξιοσημείωτο πως οι υλοποιήσεις των μοντέλων Linear Discriminant Analysis και Logistic Regression έχουν συγκρίσιμη απόδοση με εκείνη του Random Forest, καθώς έχουν F1 score ίσο με 0.06.

2. Πληρότητα Προϋποθέσεων

Ένα διαφορετικό κριτήριο αξιολόγησης των υλοποιημένων μοντέλων είναι η δυνατότητά τους να εκτιμούν τουλάχιστον το 60% των εταιρειών που θα πτωχεύσουν και τουλάχιστον το 70% των εταιρειών που δε θα πτωχεύσουν. Για τον εντοπισμό των μοντέλων που πληρούν τις παραπάνω προϋποθέσεις θα αξιοποιήσουμε τις τιμές των μετρικών recall και specificity στο test set, αντίστοιχα, τις οποίες αναλύσαμε στο τμήμα «*Πειραματικά Αποτελέσματα*» της αναφοράς.

Σε αντίθεση με τα αποτελέσματα του training set, η αποδοτικότητα των μοντέλων στο training set είναι χαμηλότερη. Για την ακρίβεια, όλα τα μοντέλα έχουν τιμή specificity μεγαλύτερη του 0.8, το οποίο σημαίνει πως κάθε μοντέλο εντοπίζει επιτυχώς τουλάχιστον το 80% των υγιών εταιρειών. Από την άλλη, τα αποτελέσματα της μετρικής recall δεν είναι εξίσου ικανοποιητικά. Στο συγκεκριμένο σύνολο δεδομένων, κανένα από τα υλοποιημένα μοντέλα δεν είναι ικανό να εντοπίζει περισσότερο από το 55% των εταιρειών που θα πτωχεύσουν.

3. Βελτίωση Αποτελεσμάτων

Υπάρχουν αρκετοί τρόποι για τη βελτίωση των επιδόσεων των παραπάνω μοντέλων. Μία βασική είναι το hyperparameter tuning με χρήση τεχνικών, όπως random search και grid search. Ενδιαφέρον θα είχε, επίσης, να εξεταστεί η αποτελεσματικότητα δημιουργίας και χρήσης synthetic data σε μια προσπάθεια εξισορρόπησης των διαθέσιμων κλάσεων και αύξησης των διαθέσιμων δειγμάτων της κλάσης χρεωκοπημένων εταιρειών. Συμπληρωματικά, θα μπορούσε να εφαρμοστεί διαφορετικού είδους κανονικοποίησης των δεδομένων και να υλοποιηθούν επιπλέον μοντέλα μηχανικής μάθησης.