



**Πανεπιστήμιο Μακεδονίας
Π.Μ.Σ. στην Εφαρμοσμένη Πληροφορική (ειδίκευση Ανάπτυξη
Λογισμικού και Νέφος)
Τμήμα Εφαρμοσμένης Πληροφορικής**

ΜΕΘΟΔΟΙ ΚΑΙ ΕΡΓΑΛΕΙΑ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

1η Εργασία (Επιβλεπόμενη μάθηση – Ταξινόμηση)

Φοιτητής: Καλαϊτζίδης Δημήτριος mai24018

Εισαγωγή

Στο πλαίσιο αυτής της εργασίας, αναλύουμε ένα σύνολο δεδομένων που παρέχονται έτοιμα από έναν οργανισμό, τα οποία περιλαμβάνουν ποικίλες μετρήσεις απόδοσης και δραστηριότητας εταιρειών, καθώς και την κατάστασή τους σε συγκεκριμένα έτη. Ο στόχος μας είναι να αναπτύξουμε μοντέλα επιβλεπόμενης μάθησης που θα προβλέπουν με ακρίβεια τις επιχειρήσεις που ενδέχεται να κηρύξουν χρεωκοπία.

Συγκεκριμένα, επιδιώκουμε:

- Την εξέταση και σύγκριση διαφορετικών τεχνικών ταξινόμησης για την αντιμετώπιση του προβλήματος.
- Την αξιολόγηση της απόδοσης των μοντέλων βάσει σημαντικών μετρικών, όπως το F1 score και το AUC ROC.
- Την παρουσίαση και ερμηνεία των αποτελεσμάτων μας, προσφέροντας παράλληλα πιθανές βελτιώσεις και μελλοντικές κατευθύνσεις.

Η παρούσα αναφορά παρέχει μια λεπτομερή ανάλυση της μεθοδολογίας που χρησιμοποιήθηκε, των δεδομένων που εξετάστηκαν, καθώς και των αποτελεσμάτων που προέκυψαν από την εφαρμογή των μοντέλων ταξινόμησης.

Μεθοδολογία

1. Φόρτωση των Δεδομένων:

Αρχικά, φορτώνουμε το dataset "Dataset2Use_Assignment2.xlsx" στο περιβάλλον εκτέλεσής μας.

2. Ανάμειξη των Δεδομένων:

Καθώς η αρχική διάταξη των δεδομένων μπορεί να επηρεάσει την εκπαίδευση των μοντέλων μας, προτιμούμε να ανακατέψουμε τα δεδομένα για να μειώσουμε τυχόν μεροληπτική συμπεριφορά.

3. Μετατροπή του Target:

Το στοιχείο που προβλέπουμε είναι αν μια επιχείρηση θα κηρύξει χρεωκοπία ή όχι. Έτσι, αντικαθιστούμε τις τιμές στη στήλη του στόχου:

- Τιμή 2 (που υποδηλώνει κηρυγμένη χρεωκοπία) με την τιμή 1.
- Τιμή 1 (που υποδηλώνει υγιή επιχείρηση) με την τιμή 0.

```
[ ] # Read dataset excel file from Drive
file_path = '/content/drive/MyDrive/Dataset2Use_Assignment1.xlsx'
df = pd.read_excel(file_path)

# Shuffle Dataframe
df = df.sample(frac=1).reset_index(drop=True)
# Replace values (1 to 0 and 2 to 1)
df['ΕΝΔΕΙΞΗ ΑΣΥΝΕΠΕΙΑΣ (=2) (v+1)'] = df['ΕΝΔΕΙΞΗ ΑΣΥΝΕΠΕΙΑΣ (=2) (v+1)'].replace({1: 0, 2: 1})

# Display the first few rows of the DataFrame
df.head()
```

4. Αφαίρεση της Στήλης του Έτους:

Η στήλη που περιέχει το έτος δεν προσφέρει σημαντική πληροφορία για την ταξινόμηση των επιχειρήσεων. Επομένως, αφαιρούμε αυτήν τη στήλη από τα δεδομένα.

5. Κανονικοποίηση των Τιμών:

Εφαρμόζουμε την τεχνική κανονικοποίησης στις τιμές των δεδομένων προκειμένου να τις φέρουμε στο διάστημα [0,1] με χρήση MinMaxScaler.

6. Έλεγχος για Τυχόν Ελλείπουσες Τιμές (NaN):

Ελέγχουμε τα δεδομένα για τυχόν ελλείπουσες τιμές και αναλαμβάνουμε την αντιμετώπισή τους. Αν υπάρχουν ελλείπουσες τιμές, τότε διαγράφουμε τις αντίστοιχες εγγραφές από το dataset.

Οπτικοποίηση Δεδομένων

Figure 1: Αριθμός Υγιών και Χρεοκοπημένων Επιχειρήσεων για κάθε Έτος

Για αυτό το γράφημα, μπορούμε να χρησιμοποιήσουμε ένα bar plot όπου θα απεικονίζονται ο αριθμός των υγιών και χρεοκοπημένων επιχειρήσεων για κάθε έτος.

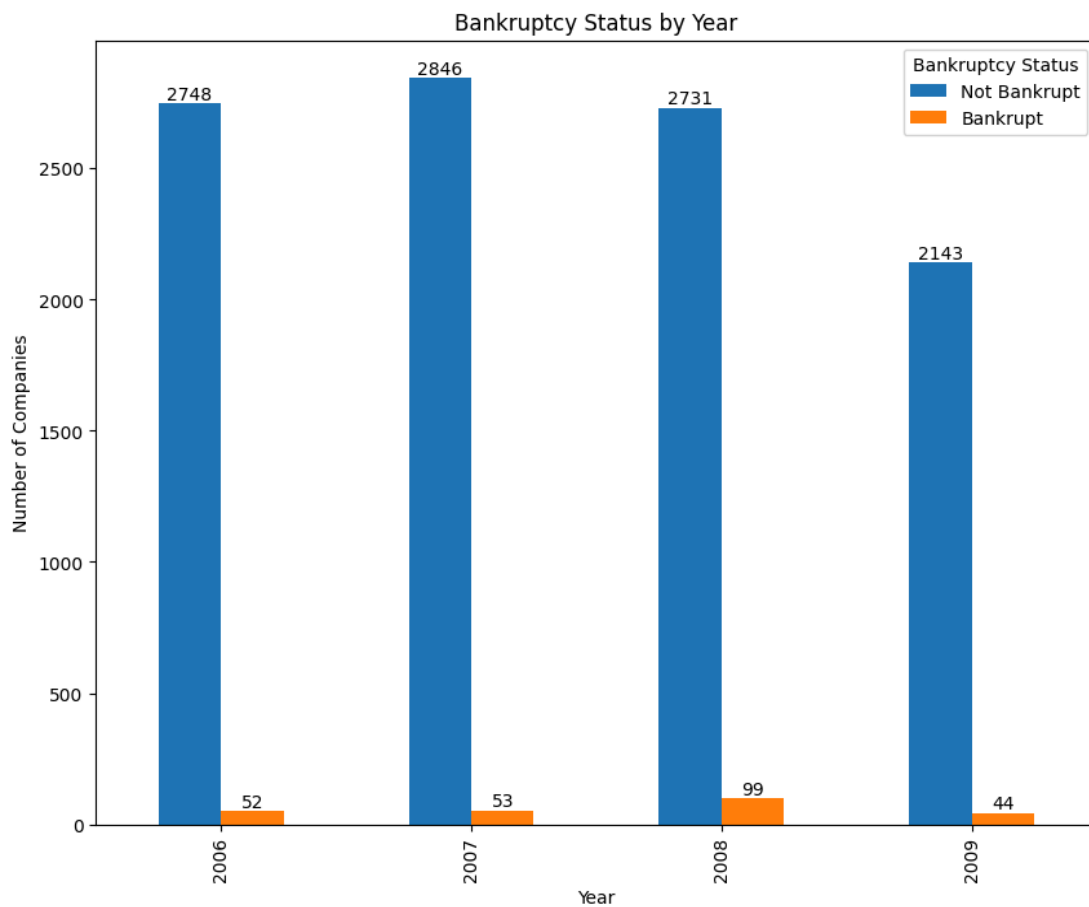
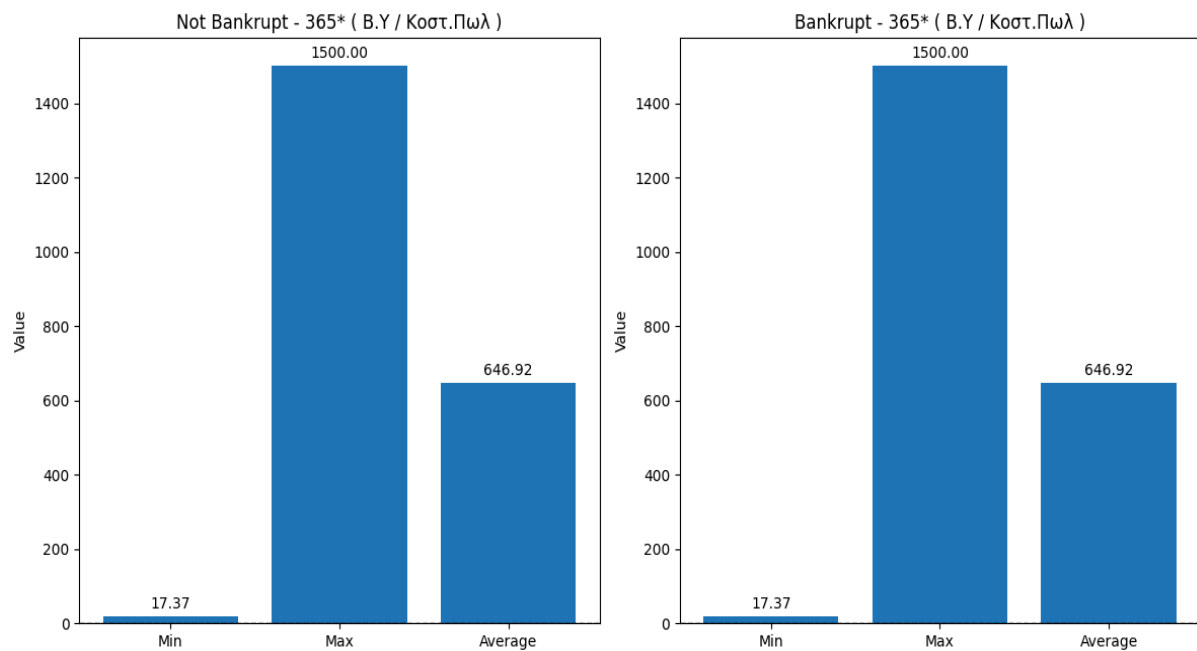


Figure 2: Ελάχιστη, Μέγιστη, Μέση Τιμή για κάθε Δείκτη

Για αυτό το γράφημα, μπορούμε να δημιουργήσουμε ένα barplot που θα παρουσιάζει την ελάχιστη, μέγιστη και μέση τιμή για κάθε δείκτη, με ένα ξεχωριστό barplot για τις υγιείς εταιρείες και ένα για τις χρεοκοπημένες.



Για τη διεξαγωγή της διαδικασίας εκπαίδευσης και αξιολόγησης των μοντέλων μας, δημιουργήσαμε 4 Stratified K-Folds. Αυτός ο τρόπος διαχωρισμού των δεδομένων επιτρέπει τη διατήρηση της ισορροπίας μεταξύ των κλάσεων χρεωκοπημένων και υγιών επιχειρήσεων σε κάθε fold.

Για κάθε fold, τυπώνουμε τον αριθμό των χρεωκοπημένων και υγιών επιχειρήσεων στο Train και Test Set:

```
Fold 1:
Healthy companies - Train: 7851, Test: 2617
Bankrupt companies - Train: 186, Test: 62

Fold 2:
Healthy companies - Train: 7851, Test: 2617
Bankrupt companies - Train: 186, Test: 62

Fold 3:
Healthy companies - Train: 7851, Test: 2617
Bankrupt companies - Train: 186, Test: 62

Fold 4:
Healthy companies - Train: 7851, Test: 2617
Bankrupt companies - Train: 186, Test: 62

Overall results:
Average number of healthy companies in train set: 7851.0
Average number of bankrupt companies in train set: 186.0
Average number of healthy companies in test set: 2617.0
Average number of bankrupt companies in test set: 62.0
```

Προετοιμασία Μοντέλων

Η προετοιμασία των μοντέλων περιλαμβάνει τη διαδικασία της εισαγωγής και της υπερπαραμετροποίησης τους. Αρχικά, εισάγουμε τις αντίστοιχες βιβλιοθήκες της βιβλιοθήκης Scikit-learn για τα μοντέλα που θα χρησιμοποιήσουμε. Έπειτα, ορίζουμε τους υπερπαραμετρικούς χώρους για κάθε μοντέλο, διαμορφώνοντας ένα λεξικό με τα ονόματα των μοντέλων ως κλειδιά και τις παραμέτρους προς εξέταση ως τιμές.

Έπειτα, ορίζουμε τα μοντέλα χρησιμοποιώντας αντικείμενα GridSearchCV. Αυτό το βήμα εμπεριέχει τη δημιουργία ενός pipeline που περιλαμβάνει το μοντέλο που εξετάζουμε και την αντίστοιχη διαδικασία αναζήτησης υπερπαραμέτρων. Η αναζήτηση υπερπαραμέτρων πραγματοποιείται με χρήση του GridSearchCV, ενώ το σκορ που χρησιμοποιείται για την αξιολόγηση του μοντέλου είναι η recall, καθώς έχει σημασία να ανιχνεύουμε όσο το δυνατόν περισσότερες εταιρείες που θα κηρύξουν χρεωκοπία.

Εδώ είναι ένα παράδειγμα του κώδικα που χρησιμοποιήθηκε για τη δημιουργία και την υπερπαραμετροποίηση των μοντέλων:

```

param_grids = {
    "Linear Discriminant Analysis": {},
    "Logistic Regression": {
        "clf__C": [0.1, 1, 10],
        "clf__solver": ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
        "clf__class_weight": ['balanced', None]
    },
    "Decision Trees": {
        "clf__max_depth": [None, 10, 15, 20],
        "clf__min_samples_split": [2, 10, 20],
        "clf__criterion": ['gini', 'entropy'],
        "clf__min_samples_leaf": [1, 5, 10],
        "clf__min_weight_fraction_leaf": [0.0, 0.1, 0.2, 0.3, 0.4, 0.5],
        "clf__max_features": ['auto', 'sqrt', 'log2', None]
    },
    "Random Forests": {
        "clf__n_estimators": [100],
        "clf__max_depth": [None],
        "clf__criterion": ['gini'],
        "clf__min_samples_split": [2],
        "clf__min_samples_leaf": [1],
        "clf__max_features": ['sqrt']
    },
    "k-Nearest Neighbors": {
        "clf__n_neighbors": [3, 5, 7, 9],
        "clf__algorithm": ['auto', 'ball_tree', 'kd_tree', 'brute'],
        "clf__leaf_size": [10, 20, 30, 40, 50],
        "clf__weights": ['uniform', 'distance']
    },
    "Naïve Bayes": {}, # No hyperparameters to tune for Naïve Bayes
    "Support Vector Machines": {
        "clf__C": [0.1, 1, 10],
        "clf__kernel": ["linear", "rbf"],
        "clf__gamma": ['scale', 'auto'],
        "clf__class_weight": ['balanced', None]
    },
}

```

```
# Define models with pipelines and GridSearchCV for hyperparameter tuning
```

```
models = {  
    "Linear Discriminant Analysis": GridSearchCV(  
        Pipeline([('clf', LinearDiscriminantAnalysis())]),  
        param_grids["Linear Discriminant Analysis"],  
        cv=stratified_kfold, scoring='recall', n_jobs=-1  
    ),  
    "Logistic Regression": GridSearchCV(  
        Pipeline([('clf', LogisticRegression())]),  
        param_grids["Logistic Regression"],  
        cv=stratified_kfold, scoring='recall', n_jobs=-1  
    ),  
    "Decision Trees": GridSearchCV(  
        Pipeline([('clf', DecisionTreeClassifier())]),  
        param_grids["Decision Trees"],  
        cv=stratified_kfold, scoring='recall', n_jobs=-1  
    ),  
    "Random Forests": GridSearchCV(  
        Pipeline([('clf', RandomForestClassifier())]),  
        param_grids["Random Forests"],  
        cv=stratified_kfold, scoring='recall', n_jobs=-1  
    ),  
    "k-Nearest Neighbors": GridSearchCV(  
        Pipeline([('clf', KNeighborsClassifier())]),  
        param_grids["k-Nearest Neighbors"],  
        cv=stratified_kfold, scoring='recall', n_jobs=-1  
    ),  
    "Naïve Bayes": GridSearchCV(  
        Pipeline([('clf', NaiveBayesClassifier())]),  
        param_grids["Naïve Bayes"],  
        cv=stratified_kfold, scoring='recall', n_jobs=-1  
    )  
}
```


Εκπαίδευση Μοντέλων και Αξιολόγηση

Στο στάδιο αυτό, κάθε μοντέλο εκπαιδεύτηκε χρησιμοποιώντας το διαχωρισμένο dataset σε κάθε φολντ του συνόλου δεδομένων. Κατόπιν, τα εκπαιδευμένα μοντέλα αξιολογήθηκαν χρησιμοποιώντας διάφορες μετρικές απόδοσης, συμπεριλαμβανομένου του **F1-score**, του **Area Under the Receiver Operating Characteristic Curve (AUC ROC)**, καθώς και του **Recall** και της **Specificity**. Μόνο τα μοντέλα που πληρούσαν συγκεκριμένα κριτήρια απόδοσης, όπως το **Recall ≥ 0.60** και η **Specificity (True Negative Rate) ≥ 0.70** , επιλέχθηκαν για περαιτέρω ανάλυση και χρήση.

Επιλέξαμε τα συγκεκριμένα κριτήρια απόδοσης (**Recall ≥ 0.60** και **Specificity ≥ 0.70**) με σκοπό να ικανοποιηθούν οι παρακάτω περιορισμοί:

Το μοντέλο πρέπει να βρίσκει με ποσοστό επιτυχίας τουλάχιστον 60% τις εταιρείες που θα πτωχεύσουν. Αυτό είναι σημαντικό για να εξασφαλίσουμε ότι το μοντέλο μας είναι αποτελεσματικό στην πρόβλεψη των πραγματικών κρίσιμων καταστάσεων, όπως η χρεωκοπία εταιρειών.

Το μοντέλο πρέπει να βρίσκει με ποσοστό επιτυχίας τουλάχιστον 70% τις εταιρείες που δεν θα πτωχεύσουν. Αυτό είναι εξίσου σημαντικό, καθώς θέλουμε να μειώσουμε τον αριθμό των λανθασμένων προβλέψεων που θα οδηγούσαν σε παραπλανητικά αποτελέσματα, διασφαλίζοντας έτσι την αξιοπιστία του μοντέλου μας στην πρόβλεψη της μη χρεωκοπίας εταιρειών.

Αξιολόγηση των μοντέλων:

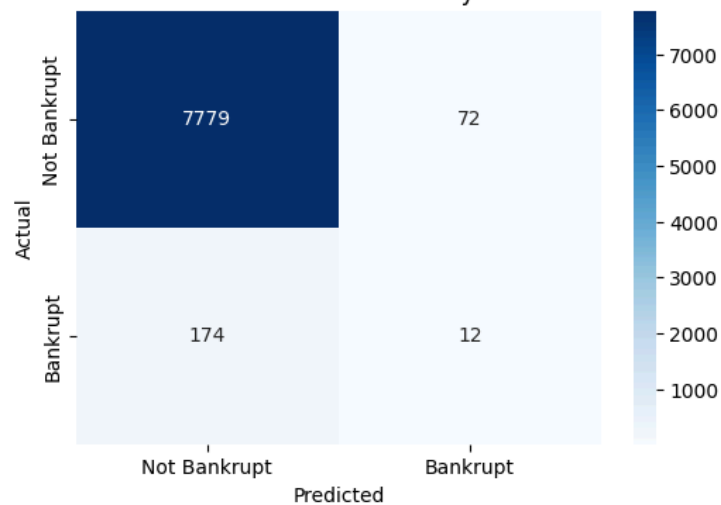
Μετά την εκπαίδευση, αξιολογούμε τα μοντέλα χρησιμοποιώντας διάφορες μετρικές απόδοσης. Αυτές περιλαμβάνουν το F1-score, το οποίο αποτελεί ένα συνδυασμό της ακρίβειας και της ανάκλησης, καθώς και το Area Under the Receiver Operating Characteristic (ROC AUC) curve, που μετρά την ικανότητα του μοντέλου να διαχωρίσει τις δύο κατηγορίες. Επιπλέον, υπολογίζουμε τα True Positives (TP), True Negatives (TN), False Positives (FP) και False Negatives (FN) για κάθε μοντέλο και για κάθε διαχωρισμό (train και test set), προκειμένου να αναλύσουμε την ακρίβεια και την απόδοσή τους.

Απεικόνιση αποτελεσμάτων:

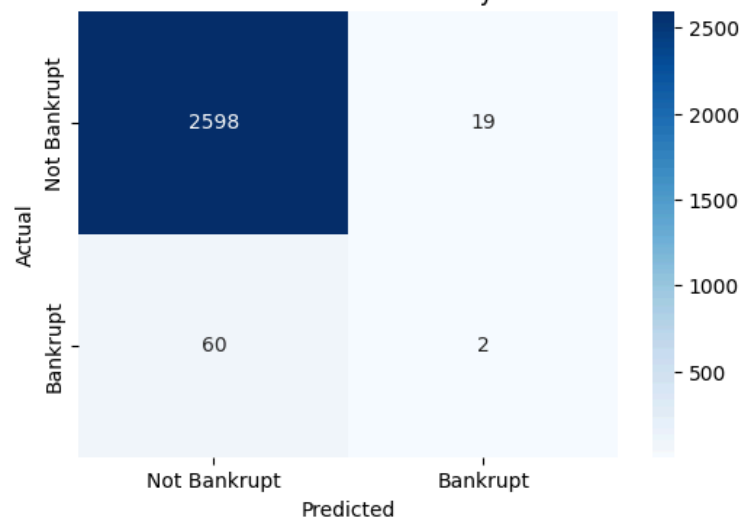
Για να έχουμε μια οπτική αναπαράσταση των αποτελεσμάτων, δημιουργούμε γραφήματα που παρουσιάζουν τους πίνακες σύγχυσης (confusion matrices) για τα train και test sets. Αυτά τα γραφήματα μας βοηθούν να κατανοήσουμε καλύτερα την απόδοση των μοντέλων και τις περιοχές όπου ενδέχεται να υπάρχουν προβλήματα κατά την πρόβλεψη.

Με βάση αυτές τις διαδικασίες, μπορούμε να προχωρήσουμε σε μια λεπτομερή ανάλυση των μοντέλων μας και των αποτελεσμάτων τους, παρέχοντας σημαντική πληροφορία για την αποτελεσματικότητά τους.

Confusion Matrix - Linear Discriminant Analysis - Train Set - Fold 1



Confusion Matrix - Linear Discriminant Analysis - Test Set - Fold 1



Αντιμετώπιση Ανισορροπίας: Αφαίρεση Υγιών Εταιρειών για Βελτίωση Απόδοσης

Στο στάδιο αυτό, λόγω της υπερβολικής ανισορροπίας στο dataset και της αδυναμίας των μοντέλων να παράγουν ικανοποιητικά αποτελέσματα, προχωρήσαμε σε μια διαδικασία αντιμετώπισης αυτού του προβλήματος. Ειδικότερα, αφαιρέσαμε τυχαία επιλεγμένες υγιείς εταιρείες από το dataset, ώστε να επιτύχουμε μια πιο ισορροπημένη αναλογία μεταξύ των υγιών και των χρεωκοπημένων εταιρειών. Η νέα αναλογία που επιτεύχθηκε ήταν τρεις υγιείς εταιρείες προς μία χρεωκοπημένη.

Με αυτόν τον τρόπο, προσπαθήσαμε να διορθώσουμε την ανισορροπία του dataset προς όφελος της απόδοσης των μοντέλων πρόβλεψης. Η επίτευξη μιας πιο ισορροπημένης αναλογίας μεταξύ των κλάσεων επέτρεψε στα μοντέλα να εκπαιδευτούν και να αξιολογηθούν με πιο αξιόπιστο τρόπο, αναμένοντας ότι θα βελτιώσει την ικανότητά τους να προβλέπουν τη χρεωκοπία εταιρειών.

Στη συνέχεια, ακολουθήσαμε την ίδια διαδικασία με πριν, δηλαδή εκπαίδευση των μοντέλων χρησιμοποιώντας το τροποποιημένο dataset και αξιολόγησή τους με τις ίδιες μετρικές απόδοσης.

Ανάλυση Απόδοσης Μοντέλων: Επίδραση του Unbalanced Dataset και Διαφορές μεταξύ Train και Test Sets

Η παρούσα ενότητα παρουσιάζει την επίδραση που έχει ένα unbalanced dataset στην απόδοση των μοντέλων και την διαφοροποίηση της απόδοσης μεταξύ train και test sets. Τα αποτελέσματα παρουσιάζονται με τη βοήθεια γραφημάτων, που αποτυπώνουν τις διαφορές και τις επιπτώσεις αυτών των παραγόντων.

Figure 1: Στο παρακάτω διάγραμμα παρατηρούμε την επίδραση που έχει ένα σύνολο δεδομένων στο οποίο η ποσότητα της μίας κατηγορίας υπερέχει κατά πολύ την ποσότητα της άλλης. Παρατηρούμε σαφώς βελτιωμένες τις μετρικές όταν εφαρμόσαμε το 3 προς 1 στο σύνολο δεδομένων μας. Αυτή η επεξεργασία οδήγησε σε μια πιο ισορροπημένη κατανομή δεδομένων και επιφέρει βελτίωση στην απόδοση των μοντέλων μας.

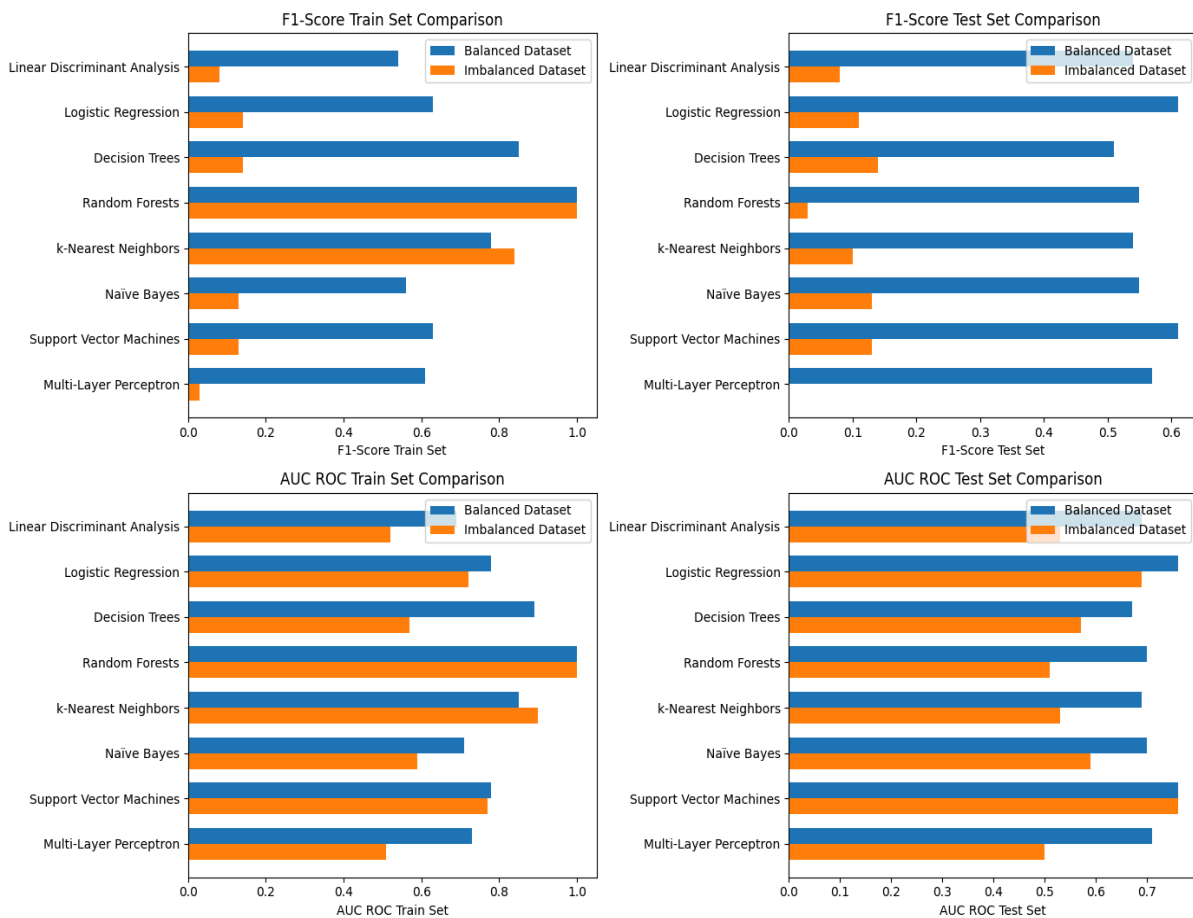
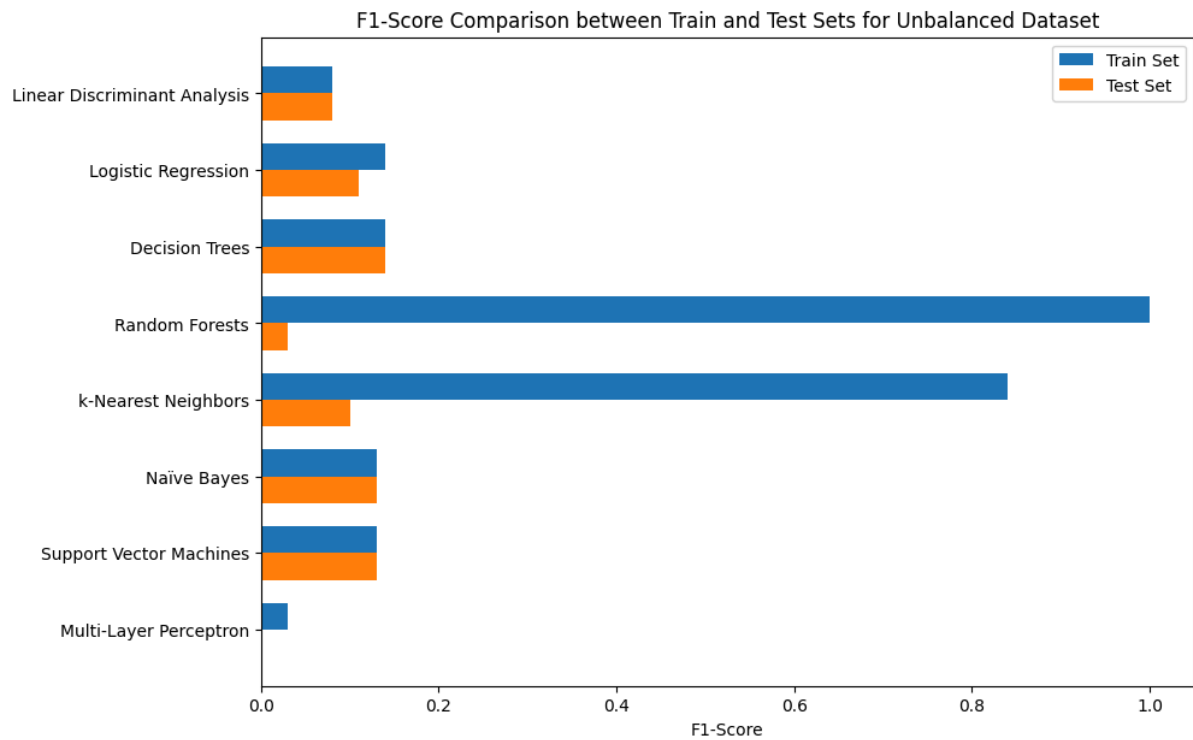
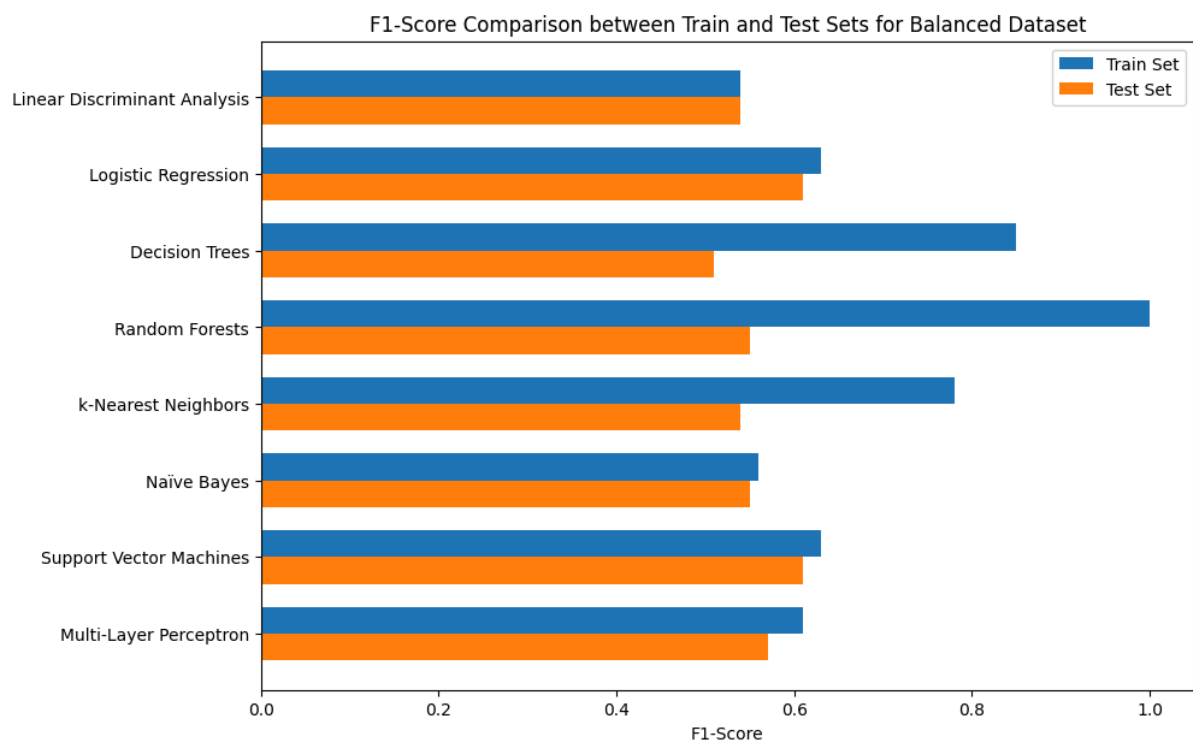
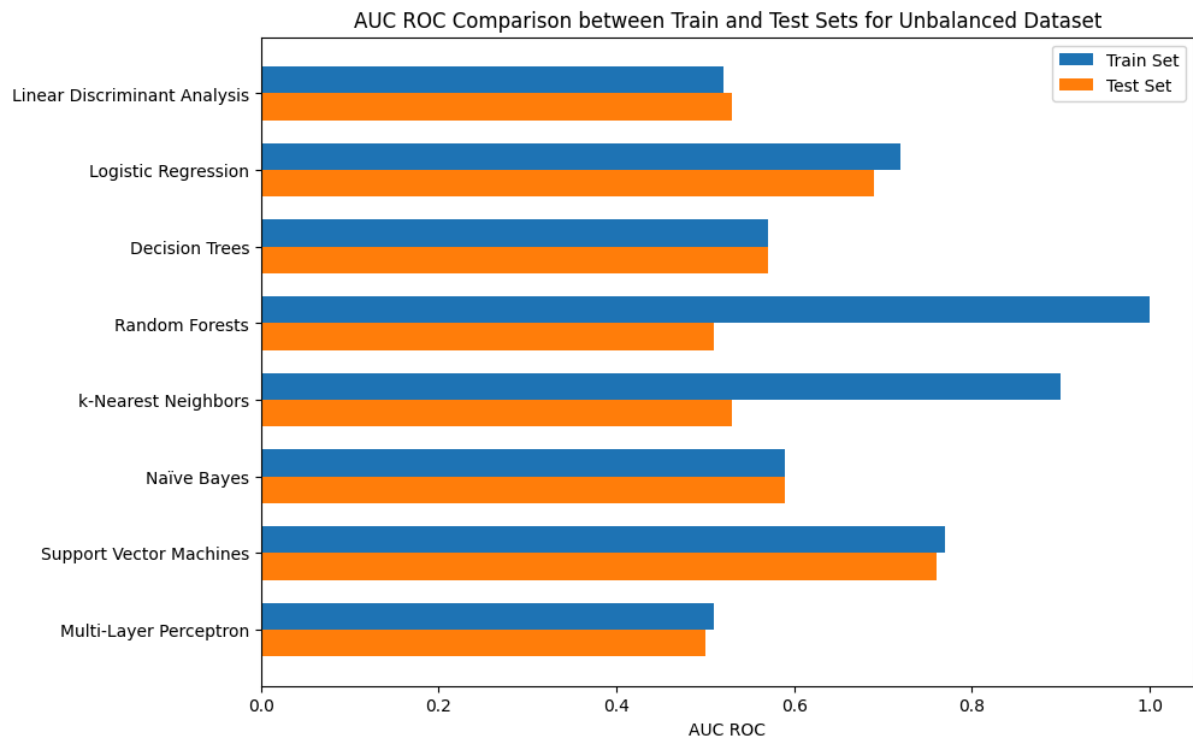
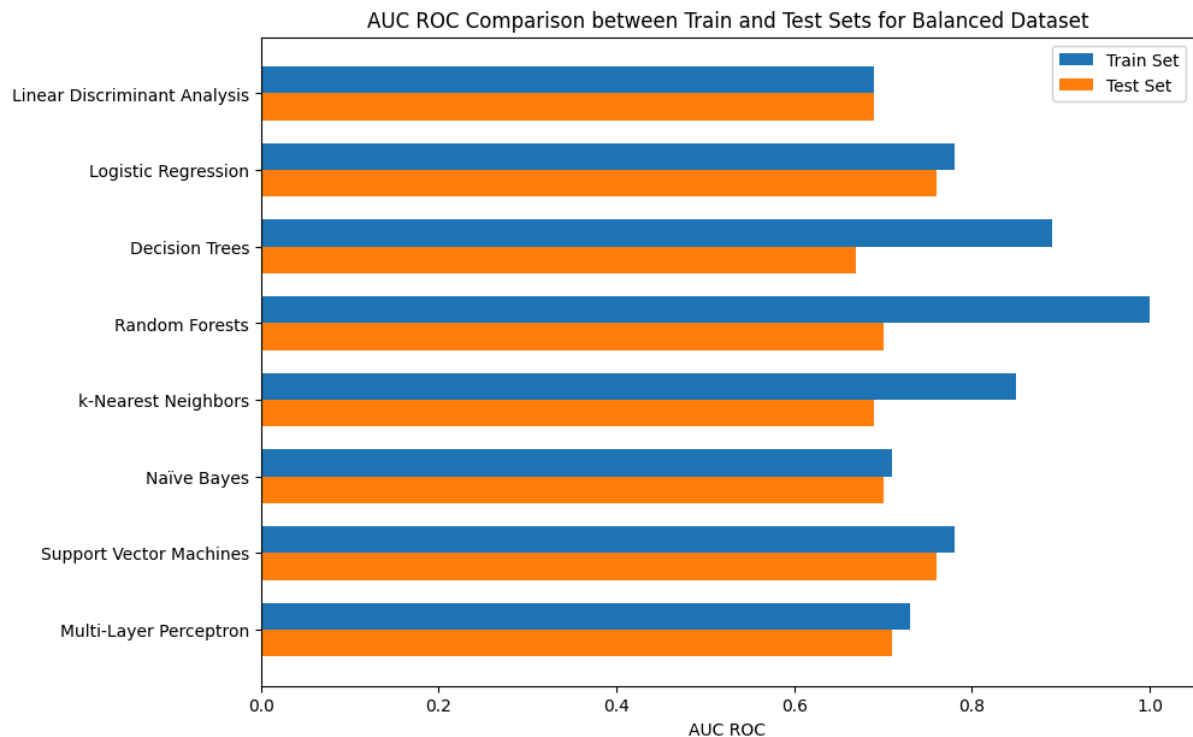


Figure 2: Στο παρακάτω διάγραμμα παρατηρούμε την διαφοροποίηση της απόδοσης μεταξύ train και test sets. Παρατηρούμε σε μερικά μοντέλα πως υπάρχει μεγάλη διαφορά στην απόδοση μεταξύ των Train και Test Set. Αυτό πιθανότατα σημαίνει overfitting, δηλαδή το μοντέλο μαθαίνει ικανοποιητικά στο σύνολο εκπαίδευσης αλλά δυσκολεύεται στην γενίκευση. Επίσης, παρατηρούμε σημαντική διαφορά στις αποδόσεις, με τις αποδόσεις στο ισορροπημένο σύνολο δεδομένων να είναι σαφώς πιο ικανοποιητικές.







Επιτυχόντα Μοντέλα

Τα μοντέλα που παρήγαγαν ικανοποιητικά αποτελέσματα στο σύνολο δεδομένων που υπάρχει ανισορροπία είναι τα εξής:

1. Support Vector Machines

- **Train Set**

- ❖ Μέσο F1 Score: 0.13
- ❖ Μέσο AUC ROC: 0.77
- ❖ Μέσο Recall: 0.79
- ❖ Μέσο Specificity: 0.74

- **Test Set**

- ❖ Μέσο F1 Score: 0.13
- ❖ Μέσο AUC ROC: 0.76
- ❖ Μέσο Recall: 0.77
- ❖ Μέσο Specificity: 0.75

- **Βέλτιστες Παράμετροι:** {'clf__C': 1, 'clf__class_weight': 'balanced', 'clf__gamma': 'auto', 'clf__kernel': 'rbf'}

Αυτά τα αποτελέσματα προέκυψαν από την αξιολόγηση των μοντέλων στο dataset, το οποίο χαρακτηρίζεται από ανισορροπία, με το ποσοστό των υγιών εταιρειών να είναι σημαντικά υψηλότερο από το ποσοστό των χρεωκοπημένων.

Το μοντέλο Support Vector Machines δείχνει ικανοποιητικό recall και specificity, αλλά το χαμηλό F1 Score υποδεικνύει προβλήματα γενίκευσης και ακρίβειας. Αυτό οφείλεται κυρίως στην ανισορροπία των δεδομένων, καθώς και στην επίδραση των υπερπαραμέτρων. Παρόλο που τα αποτελέσματα σε recall και specificity είναι ικανοποιητικά, η χαμηλή ακρίβεια σημαίνει ότι το μοντέλο μπορεί να κάνει λανθασμένες προβλέψεις. Αυτό επισημαίνει την ανάγκη για περαιτέρω βελτιστοποίηση και εξέταση των παραμέτρων του μοντέλου.

Με την εξισορρόπηση του dataset, παρατηρήσαμε σημαντική βελτίωση στην ικανότητα πρόβλεψης των μοντέλων. Συγκεκριμένα, τα μοντέλα Logistic Regression και Support Vector Machines έδειξαν σημαντική αύξηση της ανάκλησης (Recall) και της ειδικότητας (Specificity). Τα αποτελέσματα αυτών των δύο μοντέλων ήταν τα παρακάτω:

1. Logistic Regression:

- **Train Set**

- ❖ **Average F1 Score: 0.63**
- ❖ **Average AUC ROC: 0.78**
- ❖ **Average Recall: 0.79**
- ❖ **Average Specificity: 0.75**

- **Test Set**

- ❖ **Average F1 Score: 0.61**
- ❖ **Average AUC ROC: 0.76**
- ❖ **Average Recall: 0.77**
- ❖ **Average Specificity: 0.74**

- **Βέλτιστες Παράμετροι: {'clf__C': 0.1, 'clf__class_weight': 'balanced', 'clf__solver': 'newton-cg'}**

2. Support Vector Machines

- Train Set

- ❖ Average F1 Score: 0.63
- ❖ Average AUC ROC: 0.78
- ❖ Average Recall: 0.81
- ❖ Average Specificity: 0.73

- Test Set

- ❖ Average F1 Score: 0.61
- ❖ Average AUC ROC: 0.76
- ❖ Average Recall: 0.79
- ❖ Average Specificity: 0.73

- Βέλτιστες Παράμετροι: {'clf__C': 1, 'clf__class_weight': 'balanced', 'clf__gamma': 'auto', 'clf__kernel': 'rbf'}

Συμπεράσματα

Και τα δύο μοντέλα παρουσιάζουν παρόμοιες επιδόσεις, με μικρές διαφορές στις μετρικές ανάκλησης και ειδικότητας:

Ανάκληση (Recall): Το SVM έχει ελαφρώς υψηλότερη ανάκληση και στο train set (0.81 έναντι 0.79) και στο test set (0.79 έναντι 0.77). Αυτό σημαίνει ότι το SVM είναι ελαφρώς καλύτερο στο να εντοπίζει τις εταιρείες που θα πτωχεύσουν.

Ειδικότητα (Specificity): Το Logistic Regression έχει ελαφρώς υψηλότερη ειδικότητα στο train set (0.75 έναντι 0.73). Στο test set, και τα δύο μοντέλα έχουν την ίδια ειδικότητα (0.73). Επειδή η ανάκληση είναι ιδιαίτερα σημαντική για την πρόβλεψη πτωχεύσεων (καθώς προτιμούμε να εντοπίσουμε όσο το δυνατόν περισσότερες πτωχεύσεις), το SVM φαίνεται να είναι το καλύτερο μοντέλο, δεδομένης της ελαφρώς ανώτερης απόδοσής του στην ανάκληση και της ισοδύναμης ή ελαφρώς χαμηλότερης ειδικότητας.

F1 Score: Και τα δύο μοντέλα έχουν το ίδιο F1 Score στο train set (0.63) και στο test set (0.61), που υποδεικνύει ότι έχουν παρόμοια ισορροπία μεταξύ ανάκλησης και ακρίβειας.

AUC ROC: Και τα δύο μοντέλα έχουν το ίδιο AUC ROC στο train set (0.78) και στο test set (0.76), που δείχνει παρόμοια ικανότητα διάκρισης μεταξύ των τάξεων.

Και τα δύο μοντέλα παρουσιάζουν παρόμοιες επιδόσεις, με μικρές διαφορές στις μετρικές ανάκλησης και ειδικότητας:

Παρόλο που το Support Vector Machines (SVM) έχει ελαφρώς καλύτερο recall, και τα δύο μοντέλα παρουσιάζουν παρόμοια απόδοση στις άλλες μετρικές (F1 Score, AUC ROC, Specificity). Ωστόσο, η ελαφρώς καλύτερη απόδοση του SVM στην ανάκληση το καθιστά το καλύτερο μοντέλο για την πρόβλεψη πτωχεύσεων εταιρειών, καθώς προτιμάται η μεγιστοποίηση της ανάκλησης για τον εντοπισμό όσο το δυνατόν περισσότερων πτωχεύσεων.