ΕΡΓΑΣΙΑ ΕΞΟΡΥΞΗΣ ΔΕΔΟΜΕΝΩΝ (2024-2025)

ΕΙΡΗΝΗ ΜΑΡΙΑ ΛΥΚΟΥΔΗ ΑΜ :2021050

Α) Προετοιμασία

1. Μετατροπή μη αριθμητικών τιμών σε `NaN`:

df = df.apply(pd.to_numeric, errors='coerce')
Στο σύνοολο δεδομένων μας(test_unlabeled.csv) , υπηρχα αρκετές μη αριθμητικές τιμές και συγκεκριμένα πολλά ερωτηματικα (?) τα οποία δεν μπορούν να υποβληθούν σε μαθηματικές πράξεις ή να γίνουν αποδεκτά αντιστοιχα από τους περισσότερους αλγόριθμους μηχανικής μάθησης που κάναμε πειράματα και επειτα χρησιμοποιήσαμε

- Η χρήση της εντολής pd.to_numeric με την παράμετρο errors='coerce' επιτρέπει τη μετατροπή αυτών των τιμών σε NaN, που είναι μια ειδική τιμή της Python για την αναπαράσταση ελλιπών δεδομένων

2. Αντικατάσταση των `NaN` με τη μέση τιμή κάθε στήλης:

df = df.fillna(df.mean())
Η μέση τιμή είναι μια στατιστικά ουδέτερη επιλογή για την αντικατάσταση ελλιπών δεδομένων ετσι την χρησιμοποιήσαμε για να βεβαιωθούμε οτι δεν θα υπαρχει κενό γνώρισμα συγκεκριμένα :

- Δεν επηρεάζει σημαντικά την κατανομή της στήλης.
- Είναι λιγότερο πιθανό να προκαλέσει (overfitting) σε σχέση με άλλες στρατηγικές, όπως η χρήση μηδενικών ή μέγιστων τιμών.
- Εξασφαλίζεται συνέπεια στις τιμές κάθε στήλης, ενώ διατηρείται η δομή των δεδομένων

Μετά τη φόρτωση του αρχείου training_companydata.csv, καθαρίζουμε τα δεδομένα (X) χαρακτηριστικών με την παραπάνω μεθοδολογία για να είναι αριθμητικά απαλλαγμένα από κενά (Τα χαρακτηριστικά (X) προκύπτουν από όλες τις στήλες εκτός της τελευταίας , ενώ η τελευταία στήλη (y) X65 θεωρείται η κλάση στόχος)

Για το αρχείο test_unlabeled.csv, που δεν έχει την τελική στήλη στόχου, τα δεδομένα καθαρίζονται με την ίδια μέθοδο, εξασφαλίζοντας εξίσου την ομοιομορφία.

3. <u>Χαμηλή Συσχέτιση Μεταξύ Γνωρισμάτων</u>

- Με βάση τη διερεύνηση που κάναμε στα δεδομένα, διαπιστώθηκε ότι τα γνωρίσματα δεν παρουσίαζαν υψηλή συσχέτιση (π.χ., > 0.8), που να καθιστά κάποια από αυτά πλεονασματικά ώστε να μπορέσουμε να τα απορίψουμε. Ετσι, δεν κρίθηκε αναγκαία η αφαίρεση γνωρισμάτων λόγω συσχέτισης.

B) Κατηγοριοποίηση

Κατά τη διάρκεια της διαδικασίας εκπαίδευσης μοντέλων για πρόβλεψη η επιλογή του κατάλληλου αλγορίθμου είναι προφανές οτι παίζει καθοριστικό ρόλο στην ακρίβεια, την αποτελεσματικότητα και τη γενική απόδοση του μόντελου.Έτσι αρχικά κάναμε καποία πειράματα με διαφορους αλγορίθμους προκειμένου να καταλήξουμε στον ιδάνικο που θα εχει το βέλτιστο αποτέλεσμα και ταυτοχρόνα δεν θα παρουσιαζει overfitting .Συγκεκριμένα πειραματιστήκαμε με τους παρακάτω αλγοριθμους :

- **Logistic Regression**
- **Random Forest**
- **XGBoost**
- **Gradient Boosting**

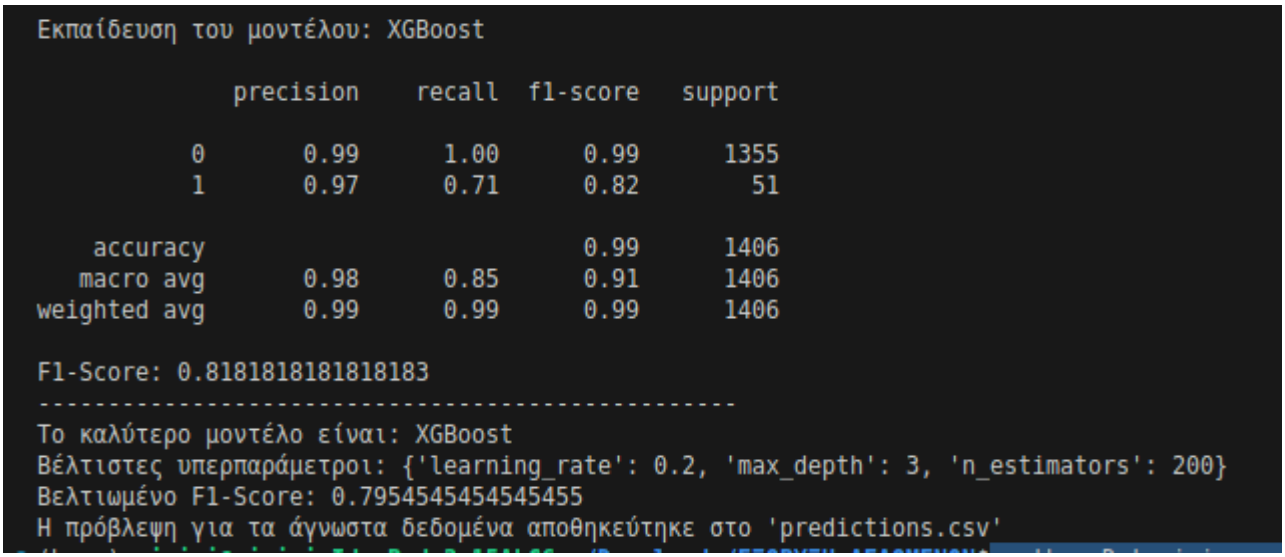
<u>Παρακάτω παρουσιάζονται τα αποτελεσματα των παραπάνω αλγορίθμων :</u>

```
Εκπαίδευση του μοντέλου: Logistic Regression
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1355
           1       0.29      0.04      0.07        51

    accuracy                           0.96      1406
   macro avg       0.63      0.52      0.52      1406
weighted avg       0.94      0.96      0.95      1406

F1-Score: 0.06896551724137931
--------------------------------------------------
Εκπαίδευση του μοντέλου: Random Forest
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1355
           1       1.00      0.47      0.64        51

    accuracy                           0.98      1406
   macro avg       0.99      0.74      0.82      1406
weighted avg       0.98      0.98      0.98      1406

F1-Score: 0.6399999999999999
--------------------------------------------------
```

*<u>Logistic Regression:</u>*Αξιολογείται με χαμηλό F1-score (0.07)για τα θετικά δειγματα που σημαινει πως η ανάκληση των θετικών δειγμάτων είναι πολύ χαμηλή.Αυτό δείχνει ότι το μοντέλο δεν καταφέρνει να αναγνωρίσει αρκετά

καλά τα θετικά δείγματα (χρεοκοπίες) γεγόνος πολύ σοβαρό για την αποδοτικότητα του μοντελου μας

*Random Forest:*Εμφανίζει πιο ικανοποιητικά αποτελέσματα με F1-score περίπου 0.64. Στα θετικά δείγματα, η ανάκληση και η ακρίβεια είναι σαφώς καλύτερες από το Logistic Regression



XGBoost: με F1-score περίπου 0.82. Παρουσιάζει αρκετά καλή ισορροπία μεταξύ precision και recall τόσο για την κλάση 0 όσο και για την κλάση 1, με υψηλή accuracy και σταθερά αποτελέσματα στα cross-validation tests ωστόσο υπαρχουν περιθώρια βελτίωσεις

```
AttributeError: 'NoneType' object has no attribute 'split'

Classification Report - Validation Set (Improved Model)
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1352
           1       0.65      0.69      0.67        54

    accuracy                           0.97      1406
   macro avg       0.82      0.84      0.83      1406
weighted avg       0.97      0.97      0.97      1406

Classification Report - Training Set
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      5404
           1       0.99      0.98      0.98      5404

    accuracy                           0.98     10808
   macro avg       0.98      0.98      0.98     10808
weighted avg       0.98      0.98      0.98     10808
```

Gradient Boosting:

Validation Set:

- Η ακρίβεια του μοντέλου είναι 0.97, με πολύ υψηλή απόδοση στη πρόβλεψη τόσο θετικών όσο και αρνητικών δειγμάτων.
- Το F1-Score είναι 0.83, που δείχνει εξαιρετική ικανότητα του μοντέλου να βρίσκει σωστά θετικά και εξίσου αρνητικά δείγματα.
- Το recall (69%) για τη χρεωκοπία είναι αρκετά ικανοποιητικό, δηλαδή το μοντέλο μπορεί να ανιχνεύει το 69% των πραγματικών χρεωκοπιών(1)

Training Set:

- Εξαιρετική απόδοση με ακρίβεια 0.98 και F1-Score 0.98.
- Το μοντέλο έχει υψηλή ακρίβεια, precision και recall σε όλα τα δείγματα, και η απόδοσή του είναι πολύ ισορροπημένη.

Συμπέρασμα: Το βελτιωμένο μοντέλο αποδίδει εξαιρετικά καλά τόσο στην εκπαίδευση όσο και στη δοκιμαστική φάση, με υψηλή ακρίβεια και πολύ καλό F1-Score.

Μετα τον πειραματισμο μας με τους παραπανω αλγοριθμους η τελική μας επιλογή ειναι ο Gradient Boosting αφού κρίθηκε από την ισορροπία μεταξύ precision, recall και F1-score, καθώς και την ικανότητά του να γενικεύει καλά για τα άγνωστα δεδομένα μας. Οι άλλοι αλγόριθμοι παρουσίασαν αρκετές αδυναμίες είτε στην ανάκτηση θετικών δειγμάτων, είτε στην ακρίβεια των προβλέψεων για θετικά δείγματα που ειναι και το σημείο που θέλουμε να δώσουμε βαση για την πρόβλεψη των ενδεχώμενων επιχειρήσεων που κινδυνέυουν απο την πτώχευση

Το Gradient Boosting έτσι αποδείχθηκε η καλύτερη επιλογή με βάση τα αποτελέσματα της πρόβλεψης και την ποιότητα της εξαγωγής στατιστικών απόδοσης. Τα cross-validation tests υποδεικνύουν σταθερά υψηλά ποσοστά ακρίβειας, γεγονός που το καθιστά το ιδανικό μοντέλο για το πρόβλημα της πρόβλεψης χρεοκοπίας.

Ανάλυση Καμπύλης Εκμάθησης

Ταυτοχρονα παραθέσαμε και την καμπύλη εκμαθησης προκειμένου οπτικοποιησουμε την εξέλιξη του μοντέλου μας

Training Score (Μπλε Γραμμή):

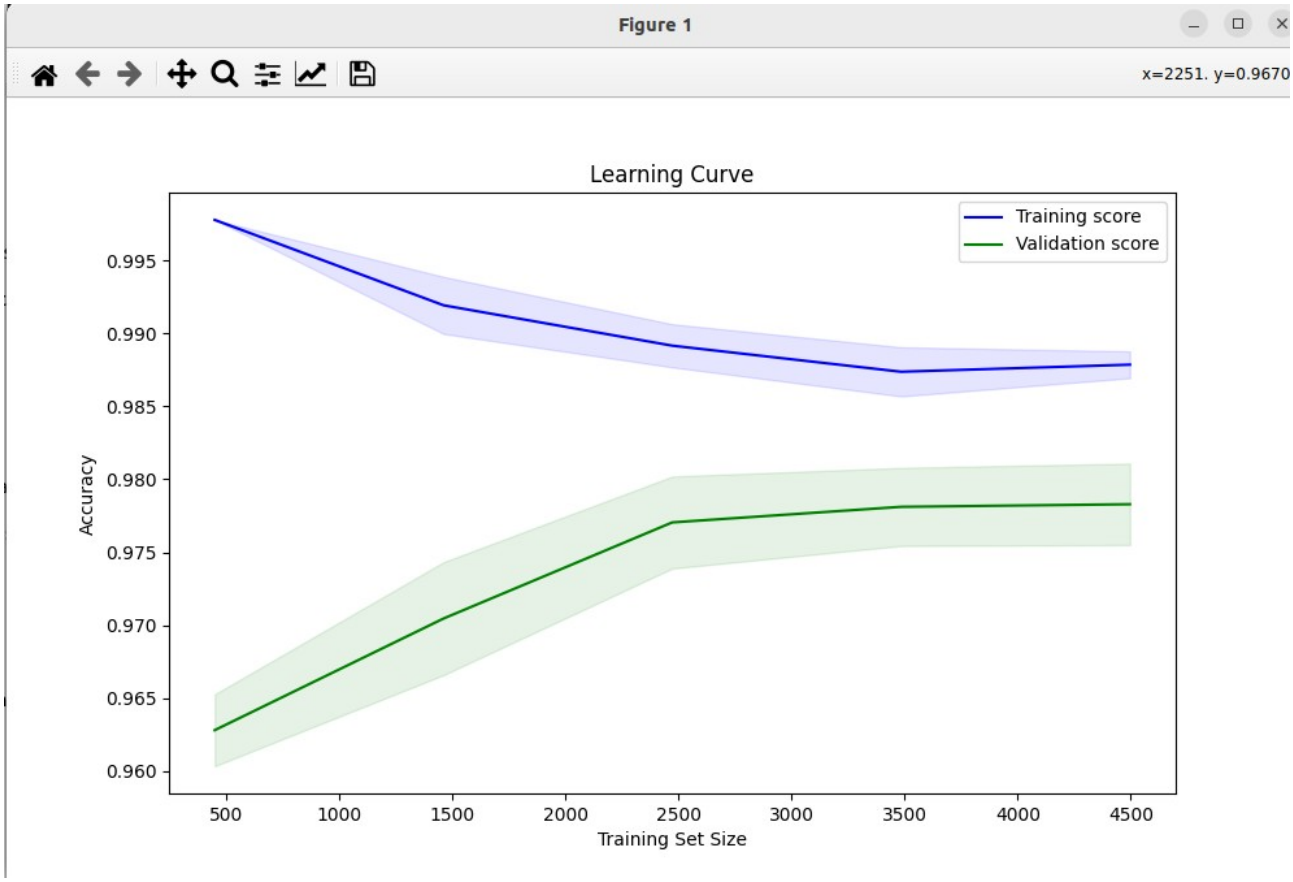
- Η ακρίβεια στο training set ξεκινάει κοντά στο **99.5%** και σταθεροποιείται ελαφρώς κάτω από το **99%** καθώς αυξάνεται το μέγεθος του συνόλου εκπαίδευσης.
- Η μικρή πτώση είναι αναμενόμενη καθώς περισσότερα δεδομένα καθιστούν την εκπαίδευση πιο δύσκολη, αποφεύγοντας το overfitting γεγονός πολύ σημαντικό για την επιλόγη του συγκεκριμένου αλγόριθμου

Validation Score (Πράσινη Γραμμή):

- Η ακρίβεια στο validation set βελτιώνεται σταδιακά όσο αυξάνεται το μέγεθος του συνόλου εκπαίδευσης.
- Ξεκινάει από περίπου **96%** και φτάνει κοντά στο **97%** με τη χρήση περισσότερων δεδομένων.
- Η διαφορά μεταξύ training και validation score (variance) μειώνεται, δείχνοντας ότι το μοντέλο γενικεύει καλά.

Interval Uncertainty (Σκίαση):

- Τα περιθώρια αβεβαιότητας (σκίαση γύρω από τις γραμμές) δείχνουν μικρή διακύμανση στα training scores, ενώ η διακύμανση στα validation scores είναι λίγο μεγαλύτερη λόγω της τυχαίας επιλογής του validation set



<u>Συμπερασματικά</u> από την καμπύλη εκμάθησης επιβεβαιώνουμε την επιλογη μας καθώς υπάρχει:

1. **Καλή Γενίκευση**:
    - Η διαφορά μεταξύ του training και validation score είναι μικρή, δείχνοντας ότι το μοντέλο αποφεύγει το overfitting όπως προαναφέρθηκε
2. **Σταδιακή Βελτίωση**:
    - Το validation score αυξάνεται συνεχώς με την προσθήκη περισσότερων δεδομένων, υποδηλώνοντας ότι το Gradient Boosting επωφελείται από μεγαλύτερα σύνολα δεδομένων.

3. **Αξιοπιστία**:

- Η σχετική σταθερότητα του training score σε συνδυασμό με τη βελτίωση του validation score δείχνει ένα καλά ισορροπημένο μοντέλο.

‘Ετσι μετά την απόφαση μας να αναπτύξουμε τον αλγόριθμο Gradient Boosting ακολουθήσαμε τις παρακάτω κινήσεις για την δημιουργία του μοντέλου μας :

- ***Χωρισμός του συνόλου για εκπαίδευση και επικύρωση*****(X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y))**

  Το 80% χρησιμοποιείται για την εκπαίδευση του μοντέλου, ενώ το 20% χρησιμοποιείται για την αξιολόγηση της απόδοσης του μοντέλου.Αυτός ο διαχωρισμός θεωρήσαμε ότι επιτρέπει αρκετά δεδομένα να χρησιμοποιηθούν για τη δημιουργία του μοντέλου ενώ παράλληλα διασφαλίζει ότι υπάρχει ένα ικανό ποσό δεδομένων επικύρωσης για να εκτιμηθεί η απόδοση με πραγματικά δεδομένα .

  Το random_state ρυθμίζεται για να διασφαλιστεί ότι τα αποτελέσματα που λαμβάνουμε είναι σταθερά και επαναλαμβανόμενα, ακόμα και αν εκτελέσουμε ξανά τον ίδιο κώδικα.

  Στη ρύθμιση του train_test_split, το stratify=y προστίθεται για να διασφαλίσουμε ότι το διαχωρισμένο training και validation data διατηρεί το ίδιο ποσοστό της κλάσης στόχου

- **Oversampling με SMOTE για αντιμετώπιση ανισορροπίας των τάξεων (smote = SMOTE(random_state=42) X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train))**

  Η SMOTE δημιουργεί συνθετικά δείγματα για να αντιμετωπίσει την ανισορροπία στις τάξεις (θετικά και αρνητικά δείγματα).Ετσι,Κανουμε χρηση του random_state=42 προκειμένου να διασφαλίσει ότι τα συνθετικά δείγματα που δημιουργούνται κάθε φορά είναι τα ίδια, κάτι που είναι χρήσιμο για την αξιολόγηση της απόδοσης του μοντέλου σε επαναληπτικά πειράματα ή για συγκρίσεις

- **Κλιμάκωση δεδομένων**(scaler = StandardScaler() X_train_resampled = scaler.fit_transform(X_train_resampled) X_val = scaler.transform(X_val)test_data_scaled = scaler.transform(test_data))

Η κλιμάκωση δεδομένων είναι απαραίτητη σε μοντέλα όπως το Gradient Boosting για να εξασφαλιστεί ότι όλα τα χαρακτηριστικά επηρεάζουν το μοντέλο ισότιμα, χωρίς να προκαλείται υπεροχή σε εκείνα με μεγαλύτερο εύρος τιμών. Χρησιμοποιώντας τον StandardScaler, εξασφαλίζουμε ομαλή και ακριβή εκπαίδευση, βελτιώνοντας έτσι τη γενίκευση του μοντέλου

- **Εκπαίδευση μοντέλου Gradient Boosting**(**gb_model = GradientBoostingClassifier(random_state=42, n_estimators=1000, max_depth=4, learning_rate=0.01, subsample=0.9, min_samples_split=100, min_samples_leaf=50, max_features="sqrt") gb_model.fit(X_train_resampled, y_train_resampled))**
- n_estimators=1000 – >Μεγάλος αριθμός δέντρων για καλή προσέγγιση.
- max_depth=4 – >Επαρκές βάθος δέντρων, αποφεύγοντας το overfitting
- learning_rate=0.01 – >Μικρό learning rate για σταθερή εκπαίδευσην ώστε τα δέντρα να μαθαίνουν με πιο αργό ρυθμό, αποτρέποντας επίσης το overfitting ενώ ταυτόχρονα με ένα μικρό learning rate, το μοντέλο έχει την ευκαιρία να βελτιώνεται σταδιακά με κάθε επιπλέον δέντρο.
- subsample=0.9 – >Υποδειγματοληψία για αποφυγή υπερπροσαρμογής.
- min_samples_split=100 –> Ελάχιστος αριθμός δειγμάτων που απαιτούνται για διάσπαση
- min_samples_leaf=50 – >Ελάχιστα δείγματα ανά φύλλο.
- max_features="sqrt" – >Χρησιμοποίηση τυχαίου υποσυνόλου χαρακτηριστικών

- **Αξιολόγηση του μοντέλου**(**y_val_pred_gb = gb_model.predict(X_val) print("\nClassification Report - Validation Set (Improved Model)")print(classification_report(y_val, y_val_pred_gb)))**

Αξιολογώντας την απόδοση του μοντέλου στο τμήμα επικύρωσης. Παρουσιάζονται πληροφορίες για την ακρίβεια, το recall, το precision και το F1-Score, που είναι πολύ κρίσιμα μέτρα για την κατανόηση της ικανότητας του μοντέλου να προβλέπει τις χρεοκοπίες

- **Αξιολόγηση στο training set(y_train_pred_gb = gb_model.predict(X_train_resampled) print("Classification Report - Training Set") print(classification_report(y_train_resampled, y_train_pred_gb)))**

  Επιβεβαίωση της απόδοσης του μοντέλου στο σύνολο εκπαίδευσης.Παρουσιάζοντας τν βαθμο υπερπροσαρμογής ή ισορροπημένης απόδοσης.

- **Καμπύλη Εκμάθησης(train_sizes, train_scores, val_scores = learning_curve(gb_model, X_train, y_train, cv=5, scoring="accuracy", n_jobs=-1))**

  Με την οπτικοποίηση της καμπύλης εκμαθησης αντιλαμβανόμαστε το πώς εξελίσσεται η απόδοση του μοντέλου καθώς αυξάνεται το μέγεθος της εκπαίδευσης

- **Διασταυρούμενη επικύρωση (Cross-validation)(cv_scores = cross_val_score(gb_model, X_train, y_train, cv=5, scoring="accuracy")print("Cross-validation scores:", cv_scores)print("Mean CV accuracy:", np.mean(cv_scores)))**

  Επιβεβαίωση της αξιοπιστίας της απόδοσης του μοντέλου μέσω διασταυρούμενης επικύρωσης.]Παρέχοντας πιο αξιόπιστες εκτιμήσεις της απόδοσης, λαμβάνοντας υπόψη διαφορετικά σύνολα δεδομένων για εκπαίδευση.

- **Προβλέψεις για το test set(predictions_full = gb_model.predict(test_data_scaled) predictions_df_full = pd.DataFrame(predictions_full, columns=["Prediction"]) predictions_df_full.to_csv("predictions.csv", index=False))**

  Πρόβλεψη για το σύνολο δοκιμών και αποθήκευση των αποτελεσμάτων στο πρώτο αρχείο με τισ προβλέψεις (predictions.csv)

- **Πρόβλεψη πιθανότητας πτώχευσης (probabilities_full = gb_model.predict_proba(test_data_scaled)[:, 1])**

  Το μοντέλο Gradient Boosting εκτελείται πάνω στο test_data_scaled, το οποίο είναι το σύνολο δοκιμών που προέρχεται από την κλιμακωμένη διαδικασία. Το predict_proba: Επιστρέφει τις πιθανότητες πρόβλεψης για κάθε δείγμα. Στο συγκεκριμένο κομμάτι, πάιρνουμε μόνο την πιθανότητα χρεοκοπίας από το δεύτερο στοιχείο κάθε γραμμής πρόβλεψης ([:, 1]),

επειδή η πρόβλεψη είναι σε δύο κατηγορίες (0 - μη χρεοκοπία, 1 – χρεοκοπία).

- **<u>Δημιουργία DataFrame με RowID και Risk_Probability</u>(results_df = pd.DataFrame({ "RowID": range(1, len(test_data) + 1), # RowID ξεκινά από 1 "Risk_Probability": probabilities_full }))**

  Δημιουργείται ένα DataFrame (results_df) που περιλαμβάνει δύο στήλες:
- RowID: Μια μοναδική ταυτότητα για κάθε δείγμα, ξεκινώντας από το 1 μέχρι το πλήθος των εταιρειών στο test_data.
- Risk_Probability: Οι πιθανότητες χρεοκοπίας που προέβλεψε το μοντέλο για κάθε εταιρεία.

  Επειτα το results_df ταξινομείται με βάση τις πιθανότητες χρεοκοπίας κατά φθίνουσα τάξη (δηλαδή από τις μεγαλύτερες προς τις μικρότερες).

  .head(50): Επιλέγει τις πρώτες 50 γραμμές από την ταξινόμηση, δηλαδή τις 50 εταιρείες με τον υψηλότερο κίνδυνο χρεοκοπίας. Και δημιουργειται το αρχειο top_50_high_risk.csv που χρειαζόμαστε

<u>Γ) Αξιολόγηση Γνωρισμάτων - Παλινδρόμηση</u>

- **<u>Εύρεση 10 κορυφαίων χαρακτηριστικών</u>(feature_importances = gb_model.feature_importances_features = X.columns sorted_idx = np.argsort(feature_importances)[::-1] top_features = features[sorted_idx][:10] print("\ nTop 10 features:", top_features))**

  Ανακάλυψη των σημαντικότερων χαρακτηριστικών που επηρεάζουν τις προβλέψεις.

- **Εκπαίδευση με μόνο τα 10 κορυφαία χαρακτηριστικά**

  Ακολουθούμε τον ιδιο τρόπο εκπαιδευσης με παραπάνω

- **<u>Αξιολόγηση του μοντέλου με κορυφαία χαρακτηριστικά</u> (Cross-validation)(y_val_pred_gb_top = gb_model_top.predict(X_val_top) print("\ nClassification Report - Validation Set (Top Features)") print(classification_report(y_val_top, y_val_pred_gb_top)))**

  Αξιολγούμε την απόδοση του μοντέλου με περιορισμένα, όμως σημαντικά χαρακτηριστικά για να επιβεβαιώσουμε την απόδοση όταν περιορίζουμε τα χαρακτηριστικά.

<u>ΣΥΓΚΡΙΣΗ ΜΕΤΑΞΥ ΤΟΥ ΤΕΛΙΚΟΥ ΒΕΛΤΙΩΜΕΝΟΥ ΜΟΝΤΕΛΟΥ ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΕΙ ΟΛΑ ΤΑ ΧΑΡΑΧΤΗΡΙΣΤΙΚΑ ΚΑΙ ΤΟΥ ΜΟΤΕΛΟΥ ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΕΙ ΜΟΝΟ ΤΑ 10 ΚΟΡΥΦΑΙΑ :</u>

<u>Συμπεράσματα από τη σύγκριση:</u>

1. ***<u>Ακρίβεια (Accuracy):</u>***
   - Το πλήρες μοντέλο δίνει ακρίβεια 0.97, ενώ το μοντέλο με τα κορυφαία χαρακτηριστικά έχει 0.93.
   - Συμπέρασμα: Το πλήρες μοντέλο δίνει καλύτερη ακριβή πρόβλεψη σε όλο το σύνολο δεδομένων.

2. ***<u>F1-Score:</u>***
   - Το πλήρες μοντέλο έχει F1-score 0.83, ενώ το μοντέλο με τα κορυφαία χαρακτηριστικά έχει 0.73.
   - Συμπέρασμα: Το πλήρες μοντέλο είναι πιο ισορροπημένο μεταξύ precision και recall.

3. ***<u>Recall:</u>***
   - Στο μοντέλο με τα κορυφαία χαρακτηριστικά, το recall είναι 0.83, ενώ στο πλήρες μοντέλο είναι 0.69.
   - Συμπέρασμα: Το μοντέλο με τα κορυφαία χαρακτηριστικά έχει καλύτερη ανάκτηση θετικών δειγμάτων (χρεοκοπίες).

4. ***<u>Precision:</u>***
   - Στο μοντέλο με τα κορυφαία χαρακτηριστικά, η precision είναι 0.34, ενώ στο πλήρες μοντέλο είναι 0.65.
   - Συμπέρασμα: Το πλήρες μοντέλο έχει καλύτερη ακρίβεια στα θετικά δείγματα.

Απόλυτα Συμπεράσματα:

- Πλήρες Μοντέλο: Προσφέρει μια καλύτερη συνολική απόδοση στην ακρίβεια και στη γενική πρόβλεψη. Είναι ισορροπημένο στο precision, recall και F1-score.
- Μοντέλο με τα 10 κορυφαία χαρακτηριστικά: Παράγει καλύτερο recall, εντοπίζοντας περισσότερα θετικά δειγμένα, αλλά μειώνει σημαντικά την

precision και το συνολικό F1-score, με αποτέλεσμα να κάνει πολλά ψευδώς θετικά.

ΤΕΛΙΚΟ ΒΕΛΤΙΩΜΕΝΟ ΜΟΝΤΕΛΟ ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΕΙ ΟΛΑ ΤΑ ΧΑΡΑΧΤΗΡΙΣΤΙΚΑ:

```
Classification Report - Validation Set (Improved Model)
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1352
           1       0.65      0.69      0.67        54

    accuracy                           0.97      1406
   macro avg       0.82      0.84      0.83      1406
weighted avg       0.97      0.97      0.97      1406
```

*ΤΟΥ ΜΟΤΕΛΟΥ ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΕΙ ΜΟΝΟ ΤΑ 10 ΚΟΡΥΦΑΙΑ ΧΡΑΧΤΗΡΙΣΤΙΚΑ **:***

```
Top 10 features: Index(['X27', 'X21', 'X37', 'X24', 'X6', 'X34', 'X13', 'X46', 'X16', 'X11'], dtype='object')

Classification Report - Validation Set (Top Features)
              precision    recall  f1-score   support

           0       0.99      0.94      0.96      1352
           1       0.34      0.83      0.49        54

    accuracy                           0.93      1406
   macro avg       0.67      0.88      0.73      1406
weighted avg       0.97      0.93      0.95      1406

(base) eirini@eirini-IdeaPad-3-15ALC6:~/Downloads/ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ$
```