Χαντζαρίδης Κύδων – ΑΕΜ: 64

Advanced Topics in Machine Learning – Project Report

Το dataset με το οποιο ασχολήθηκα ονομάζεται Myocardial infarction complications.

Το dataset περιέχει συνολικά 124 columns. Τα πρώτα 112 columns είναι δεδομένα που μπορούμε να χρησιμοποιήσουμε ως features, ενώ από την στήλη 113 έως και την στήλη 124 είναι τα columns που μπορούν να θεωρηθούν ως target classes. Το dataset αυτό από την σελίδα που το βρήκα ήταν στη μορφή .data, μετά από preproccessing έγινε το κατάλληλο parsing για να μπορέσω να δουλέψω πάνω σε αυτό. Έπειτα από έρευνα είδα πως το ίδιο dataset υπήρχε έτοιμο σε μορφή csv, οπότε το πήρα κατευθείαν ως csv από αυτή τη σελίδα.

Το συγκεκριμένο dataset χρησιμοποιήθηκε γιατί παρατήρησα πως τα predicted classes έχουν όλα έντονο imbalance ανάμεσα στις 2 κλάσεις.

Τα βήματα που εφάρμοσα για το συγκεκριμένο project είναι τα εξής:

- Pre-Proccessing
    1. Train-Test split
    2. Impute Missing Values
    3. Scaling data
    4. PCA for feature selection

- Apply Class Imbalance Techniques
- Make classification and compare results with plots and metrics

Pre-Proccessing

Στο κομμάτι του pre-proccesing, έπρεπε να δούμε τα columns του dataset κατά πόσο είναι πλήρες ή περιέχουν missing values. Αυτό έγινε με το παρακάτω κομμάτι κώδικα:

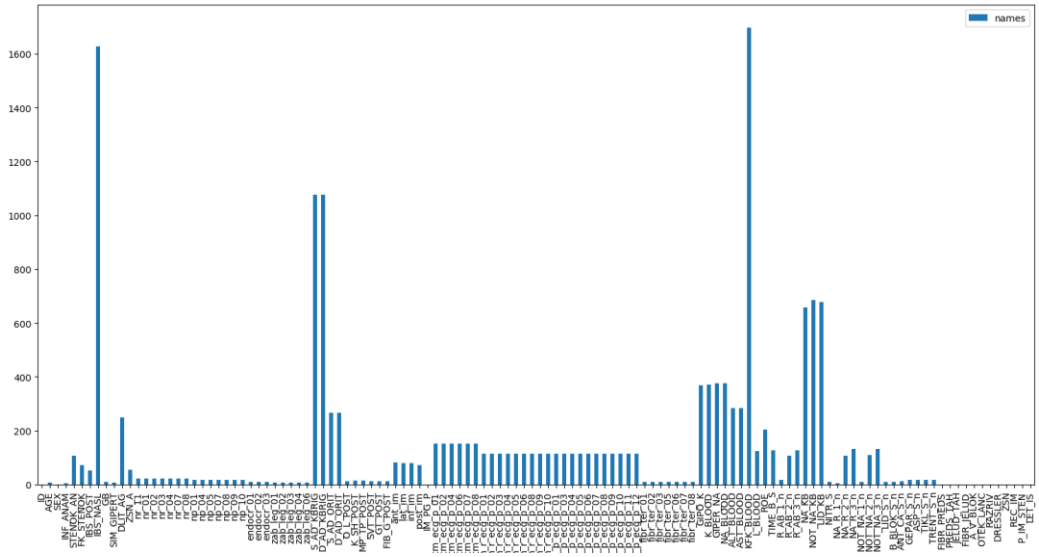
```
data.replace("?", np.nan, inplace = True)

missing_data = data.isnull()

print(data.isnull().sum())


data.isnull().sum().reset_index(name="names").plot.bar(x
='index', y='names', rot=90)
```

Οπότε πήρα το σχήμα που εμφανίζει αυτός ο κώδικας και από αυτό φαίνεται ποια columns έχουν missing values και σε τι ποσοστό.



Μέσα από αυτό το σχήμα πήρα τα columns με μεγάλο ποσοστό από missing values και τα αφαίρεσα τελείως από το dataset.

Για τα υπόλοιπα columns δοκίμασα διάφορους τρόπους για impute που οδηγούσαν όλα στην ίδια σχεδόν επίδοση.

Έπειτα χώρισα το dataset μου σε train και test με την train_test_split και στη συνέχεια έκανα impute των missing values για τα columns που παρέμειναν με την βοήθεια της συνάρτησης KNNImputer.

Εφόσον πλέον το dataset δεν έχει missing values, εφαρμόστηκε η κανονικοποίηση την δεδομένων με τη βοήθεια της συνάρτησης MinMaxScaler.

Τέλος, για να πάρουμε τα κυριότερα features εφαρμόστηκε PCA και στο imbalanced dataset αλλά και μετά τις classimbalanced techniques που θα αναλυθούν παρακάτω, για να πάρουμε τα principal components.

CLASS IMBALANCE TECHNIQUES

Η αρχική σκέψη που δηλώθηκε και στο proposal ήταν να δοκιμάσουμε σύνθεση δεδομένων με:

- SMOTE για "σύνθεση" παραδειγμάτων των κλάσεων που συναντάμε λιγότερο στο dataset. Παράλληλα με την τεχνική SMOTE θα δοκιμάσουμε και κάποιες άλλες τεχνικές, όπως την Border-Line Smote ή την μέθοδο Tomek Links για να ερευνήσουμε εάν και πόσο τα αποτελέσματα μας θα είναι καλύτερα.

- Easy Ensemble με δημιουργία τυχαίων samples από το majority class με αριθμό παραδειγμάτων ίδιο με τον αριθμό των παραδειγμάτων του minority class.

- Cluster-based sampling εφαμοργή αυτής της μεθόδου ξεχωριστά στο majority και στο minority class με σκοπό να βρούμε clusters μέσα στις κύριες κλάσεις (υποκατηγορίες) και με sampling σε αυτά τα clusters να αντιμετωπίσουμε το class imbalance.

  Κατά τη διάρκεια της υλοποίησης τα αποτελέσματα δεν ήταν ικανοποιητικά και για αυτό τον λόγο χρησιμοποιήθηκαν πιο πολλές τεχνικές, ώστε να συγκριθούν μεταξύ τους και να βγει ένα τελικό συμπέρασμα μετά την υλοποίηση όλων αυτών των τεχνικών. Οι τεχνικές που δοκιμάστηκαν είναι οι παρακάτω:

Classification in imbalanced dataset with LogisticRegression

Αρχικά δοκιμάσαμε να τρέξουμε τον αλγόριθμο του LogisticRegression για την πρόβλεψη της τιμής του column Supraventricular tachycardia (PREDS_TAH) το οποίο κατηγοριοποιεί τα δεδομένα σε ασθενείς που δεν έχουν υπερκοιλιακή ταχυκαρδία (0) και σε ασθενείς που έχουν (1). Το αρχικό dataset δεν έχει missing values σε αυτή τη κλάση και έχει τις εξής τιμές:

0: no 1680 98.82%
1: yes 20 1.18%

Το οποίο μας δείχνει πως υπάρχει έντονο class imbalance.

Τα αποτελέσματα για ένα απλό execution του Logistic Regression είναι τα εξής:

Accuracy Score : 0.974510
Balanced Accuracy Score : 0.500000
Precision Score : 0.487255
Recall Score : 0.500000
F1 Score : 0.493545
LogisticRegression: ROC AUC=0.682
LogisticRegression Unbalanced: f1=0.000 auc=0.049

Παρόλο που έχουμε υψηλό accuracy, όπως είναι αναμενόμενο, στη συγκεκριμένη περίπτωση το accuracy είναι irrelevant metric που δεν αντικατοπτρίζει την πραγματική απόδοση του μοντέλου μας. Αυτό φανερώνεται και από τις άλλες relevant μετρικές (ROC AUC, auc, Balanced Accuracy, Precision, Recall).

SMOTE / BorderLine SMOTE

Η πρώτη τεχνική αφορά την σύνθεση δεδομένων για το minority class με τη βοήθεια του SMOTE και του BorderLine SMOTE. Η ίδια ουσιαστικά τεχνική στην απλή της μέθοδο και στην μέθοδο με χρήση BorderLine.

RandomOverSampler

Η δεύτερη τεχνική αφορά το Random oversampling του minority class, ώστε τα παραδείγματα του minority class να είναι σε ίδιο αριθμό με τα παραδείγματα του majority class.

ClusterOverSampler

Σε αυτή τη περίπτωση ήθελα να δω τα αποτελέσματα ενός clustering oversampling για το minority class. Εδώ χρησιμοποίησα smote για το oversampling και Kmeans για το το κομμάτι του clustering.

Cost Sensitive Classification

Μια άλλη τεχνική που χρησιμοποιήθηκε με σκοπό να συγκριθεί με όλες τις υπόλοιπες, ενώ αρχικά ήθελα να την συνδυάσω και με το cost sensitive κομμάτι του project, είναι το classification με έναν αλγόριθμο και weights ώστε να αντιμετωπιστεί το classimbalanced. Όποτε σε αυτή τη περίπτωση έγινε classification με SVM και balanced class weight για να παρατηρηθούν τα αποτελέσματα.

UnderSampling

Η επόμενη τεχνική αφορά ένα απλό undesampling του majority class ώστε τα δεδομένα του majority class να έχουν ίδιο αριθμό με αυτά του minority. Για αυτή την τεχνική χρησιμοποιήθηκε ο αλγόριθμος NearMiss2, καθώς είχε καλύτερα αποτελέσματα από τους NearMiss1 και NearMiss3. Επίσης για την τεχνική του UnderSampling δοκιμάστηκε και ο αλγόριθμος ClusterCentroids.

Boosting Algorithm for Class Imbalance

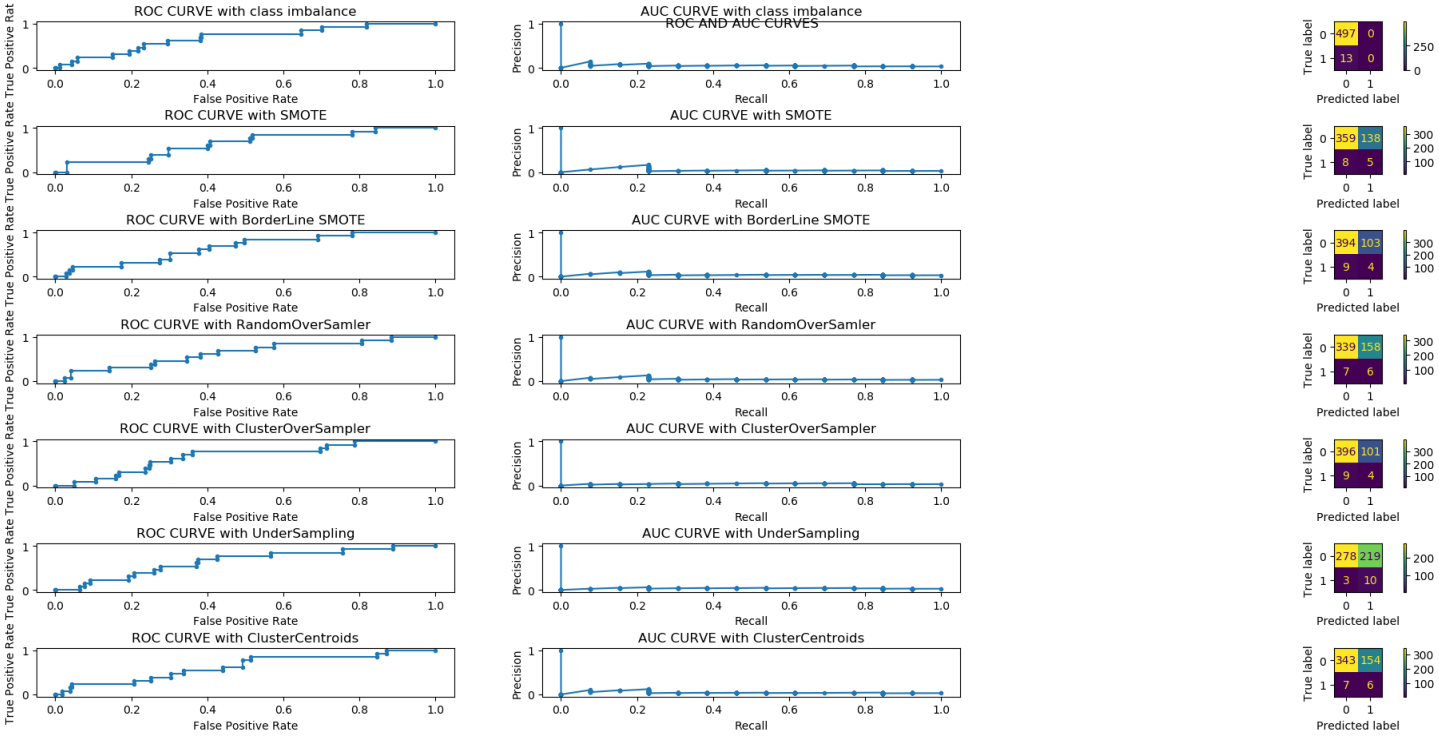
Σε αυτή τη περίπτωση χρησιμοποίησα έναν boosting αλγόριθμο για να αντιμετωπίσω το class imbalance του target class, αυτός ο αλγόριθμος είναι ο XGBClassifier. Δέχεται σαν παράμετρο την scale_pos_weight, η οποία συντονίζει τη συμπεριφορά του αλγορίθμου και μπορεί με αυτό το τρόπο να αντιμετωπίσει το class imbalance.

Σύγκριση τεχνικών

Model	Accuracy	Balanced Accuracy	Precision	Recall	F1	ROC AUC	Precision-Recall	#features (target class)
Unbalanced	**0.97**	**0.50**	**0.48**	**0.50**	**0.49**	**0.682**	**0.049**	**0: 1161, 1: 29**
SMOTE	**0.71**	**0.55**	**0.50**	**0.55**	**0.44**	**0.643**	**0.047**	**1: 1161, 0: 1161**
BordeLine SMOTE	**0.78**	**0.55**	**0.50**	**0.55**	**0.47**	**0.663**	**0.044**	**1: 1161, 0: 1161**
Random Oversampler	**0.67**	**0.57**	**0.50**	**0.57**	**0.43**	**0.638**	**0.045**	**1: 1161, 0: 1161**
Cluster Oversampler	**0.78**	**0.55**	**0.50**	**0.55**	**0.47**	**0.660**	**0.037**	**1: 1161, 0: 1161**
NearMiss2	**0.56**	**0.66**	**0.51**	**0.66**	**0.39**	**0.650**	**0.037**	**0: 29, 1: 29**
ClusterCentroids	**0.68**	**0.57**	**0.50**	**0.57**	**0.43**	**0.626**	**0.043**	**0: 29, 1: 29**

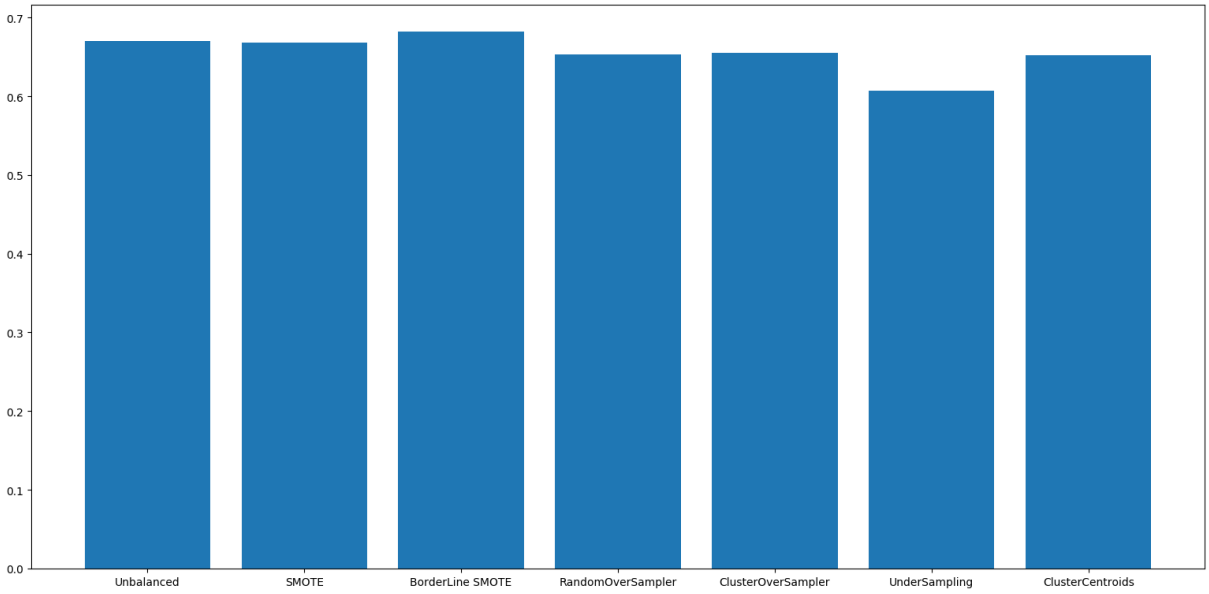
Αξίζει επίσης να σημειωθεί πως η cost sensitive προσέγγιση και ο boosting αλγόριθμος έφεραν τα καλύτερα αποτελέσματα συγκριτικά με τους υπόλοιπους. Το SVM είχε 0.72 για την ROC καμπύλη, ενώ ο boosting αλγόριθμος XGBClassifier είχε 0.70.

Παρακάτω έχουμε ένα σχήμα που μας δείχνει τις καμπύλες ROC, Precision – Recall και τον confusion matrix για κάθε περίπτωση.



Από το σχήμα καταλαβαίνουμε πως υπάρχει μια ελάχιστη βελτίωση εφαρμόζοντας τις παραπάνω τεχνικές, όμως καμία δεν έφερε τα επιθυμητά αποτελέσματα. Επίσης από το confusion matrix βλέπουμε πως οι σωστές προβλέψεις για την κλάση 0 είναι πολλές σε κάθε περίπτωση και με κάθε τεχνική, το πρόβλημα είναι στις προβλέψεις για τη κλάση 1, οπού εκεί τα αποτελέσματα δεν είναι καλά.

Άλλο ένα σχήμα που μας βοηθάει να καταλάβουμε πως δεν έχουμε την επιθυμητή βελτίωση που θα θέλαμε είναι το σχήμα που παρουσιάζει συγκεντρωτικά το roc_auc_score για κάθε περίπτωση:



Χρησιμοποιήθηκαν τόσες πολλές τεχνικές και έγιναν αλλαγές στις παραμέτρους κάθε συνάρτησης και πολλά execution με τις διαφορετικές αυτές τιμές, για να μπορέσουμε να καταλάβουμε τι φταίει και δεν έχουμε τα επιθυμητά αποτελέσματα. Κάθε τεχνική, όπως ήταν αναμενόμενο ρίχνει το accuracy του classification, όμως καμία από αυτές δεν δείχνει να βελτιώνει τις άλλες μετρικές, ενώ θα έπρεπε. Μετά από κάθε τεχνική χρησιμοποιήθηκε PCA για να ληφθούν τα 3 πρώτα principal components, που περιέχουν και τη περισσότερη πληροφορία.

Reference

- https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications
- https://leicester.figshare.com/articles/dataset/Myocardial_infarction_complications_Database/12045261
- https://machinelearningmastery.com/xgboost-for-imbalanced-classification/
- https://www.kaggle.com/ambpro/dealing-with-unbalance-eda-pca-smote-lr-svm-dt-rf
- https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets
- https://www.kdnuggets.com/2019/11/tips-class-imbalance-missing-labels.html
- **Διαφάνειες – Records – Notebooks μαθήματος.**

Χαντζαρίδης Κύδων – ΑΕΜ: 64

Advanced Topics in Machine Learning – Project Report