# Μηχανική Μάθηση
# Εργασία 1

**Μπόζας Αριστείδης**

**ΑΜ:740**

**Δερμεντζόγλου Ιωάννης**

**ΑΜ:743**

# Περιεχόμενα

## 1. Μέρος Α

Στο πρώτο μέρος της εργασίας εξετάστηκε το αντικείμενο των τεχνικών πολλαπλών μοντέλων πρόβλεψης σε συνδυασμό με το αντικείμενο της συγκριτικής αξιολόγησης μεταξύ αλγορίθμων.

Τα 10 datasets που χρησιμοποιήθηκαν από το UCI repository είναι τα εξής:

- Iris
- Wine
- Breast Cancer Wisconsin (Diagnostic)
- Balance Scale
- Hayes-Roth
- Haberman's Survival
- Liver Disorders
- Banknote Authentication
- Ionosphere
- Contraceptive Method Choice

Οι τεχνικές που χρησιμοποιήθηκαν για τα πολλαπλά μοντέλα πρόβλεψης είναι οι εξής:

- **Manipulating the training examples:** Οι τεχνικές του bagging και boosting.
- **Manipulating the target variable:** Οι τεχνικές του OnevsOne και OnevsRest.
- **Injecting randomness :** Το ensemble μοντέλο RandomForest.
- **Manipulating Features:** Τυχαία επιλογή των features και των παραδειγμάτων εκπαίδευσης με την τεχνική RandomPatches.

Όσον αφορά στα αποτελέσματα της συγκριτικής αξιολόγησης μεταξύ των αλγορίθμων που εκτελέστηκαν, αυτά παρουσιάζονται παρακάτω:

Ο πίνακας με τις τιμές της μετρικής accuracy έχει ως εξής:

| | DT | Bagged DT | AdaBoost DT | GradientBoost DT | OneVsOne DT | OneVsRest DT | RF | Bagged DT RP |
|---|---|---|---|---|---|---|---|---|
| **Iris** | 0.96 | 0.967 | 0.953 | 0.96 | 0.953 | 0.96 | 0.96 | 0.96 |
| **Wine** | 0.895 | 0.967 | 0.9 | 0.917 | 0.9 | 0.878 | 0.978 | 0.973 |
| **Breast Cancer** | 0.918 | 0.958 | 0.914 | 0.961 | 0.918 | 0.918 | 0.965 | 0.958 |
| **Balance Scale** | 0.68 | 0.679 | 0.667 | 0.705 | 0.635 | 0.743 | 0.683 | 0.602 |
| **Hayes-Roth** | 0.811 | 0.826 | 0.772 | 0.772 | 0.811 | 0.779 | 0.825 | 0.811 |
| **Haberman's Survival** | 0.626 | 0.638 | 0.569 | 0.59 | 0.619 | 0.626 | 0.671 | 0.655 |
| **Liver Disorders** | 0.567 | 0.58 | 0.568 | 0.58 | 0.567 | 0.567 | 0.571 | 0.566 |
| **Banknote Authentication** | 0.983 | 0.991 | 0.983 | 0.996 | 0.983 | 0.983 | 0.993 | 0.977 |
| **Ionosphere** | 0.881 | 0.918 | 0.879 | 0.93 | 0.881 | 0.881 | 0.938 | 0.932 |
| **Contraceptive** | 0.468 | 0.522 | 0.5 | 0.57 | 0.479 | 0.467 | 0.511 | 0.518 |

| Method Choice | | | | | | | | |
|---|---|---|---|---|---|---|---|---|

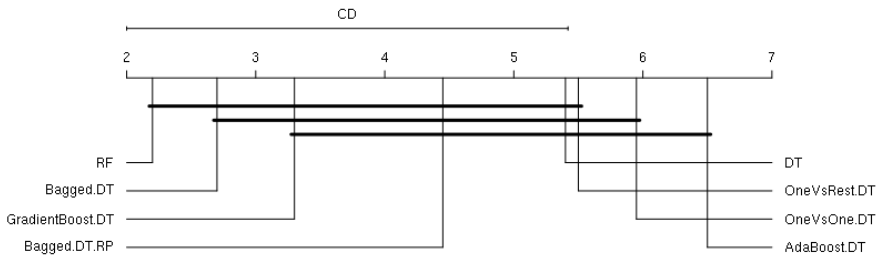
Από αυτόν τον πίνακα προκύπτει ο πίνακας με τις μέσες κατατάξεις των μοντέλων:

| | DT | Bagged DT | AdaBoost DT | GradientBoost DT | OneVsOne DT | OneVsRest DT | RF | Bagged DT RP |
|---|---|---|---|---|---|---|---|---|
| **Iris** | 4 | 1 | 7.5 | 4 | 7.5 | 4 | 4 | 4 |
| **Wine** | 7 | 3 | 5.5 | 4 | 5.5 | 8 | 1 | 2 |
| **Breast Cancer** | 6 | 3.5 | 8 | 2 | 6 | 6 | 1 | 3.5 |
| **Balance Scale** | 4 | 5 | 6 | 2 | 7 | 1 | 3 | 8 |
| **Hayes-Roth** | 4 | 1 | 7.5 | 7.5 | 4 | 6 | 2 | 4 |
| **Haberman's Survival** | 4.5 | 3 | 8 | 7 | 6 | 4.5 | 1 | 2 |
| **Liver Disorders** | 6 | 1.5 | 4 | 1.5 | 6 | 6 | 3 | 8 |
| **Banknote Authentication** | 5.5 | 3 | 5.5 | 1 | 5.5 | 5.5 | 2 | 8 |
| **Ionosphere** | 6 | 4 | 8 | 3 | 6 | 6 | 1 | 2 |
| **Contraceptive Method Choice** | 7 | 2 | 5 | 1 | 6 | 8 | 4 | 3 |
| **Average Ranks** | **5.4** | **2.7** | **6.5** | **3.3** | **5.95** | **5.5** | **2.2** | **4.45** |

Βασισμένοι στα προηγούμενα μπορούμε να κατατάξουμε τα μοντέλα από το καλύτερο στο χειρότερο ως εξής: RF, Bagged DT, GradientBoost DT, Bagged DT RP, DT, OneVsRest DT, OneVsOne DT, AdaBoost DT.

Για την στατιστική ανάλυση χρησιμοποιούμε το Iman - Davenport τεστ για την εύρεση στατιστικά σημαντικών διαφορών στην απόδοση των μοντέλων, λαμβάνουμε p-value < 0.0001 και συμπεραίνουμε ότι υπάρχουν στατιστικά σημαντικές διαφορές.

Κάνοντας χρήση του Nemenyi τεστ λαμβάνουμε αποτελέσματα που συνοψίζονται στο ακόλουθο διάγραμμα:




από το οποίο συμπεραίνουμε πως το RF έχει στατιστικά σημαντικές διαφορές από τα DT, OneVsRest DT, OneVsOne DT και AdaBoostDT και το Bagged DT από τα OneVsOne DT και AdaBoost DT. Τα συμπεράσματα αυτά επαληθεύονται και από την

εφαρμογή ενός Friedman post-hoc τεστ με διόρθωση Bergmann and Hommel των οποία τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

| | DT | Bagged.DT | AdaBoost.DT | GradientBoost.DT | OneVsOne.DT | OneVsRest.DT | RF | Bagged.DT.RP |
|---|---|---|---|---|---|---|---|---|
| DT | n/a | 0.123 | 1.000 | 0.480 | 1.000 | 1.000 | **0.045** | 1.000 |
| Bagged.DT | 0.123 | n/a | **0.011** | 1.000 | **0.045** | 0.116 | 1.000 | 0.991 |
| AdaBoost.DT | 1.000 | **0.011** | n/a | 0.056 | 1.000 | 1.000 | **0.002** | 0.797 |
| GradientBoost.DT | 0.480 | 1.000 | 0.056 | n/a | 0.171 | 0.480 | 1.000 | 1.000 |
| OneVsOne.DT | 1.000 | **0.045** | 1.000 | 0.171 | n/a | 1.000 | **0.013** | 1.000 |
| OneVsRest.DT | 1.000 | 0.116 | 1.000 | 0.480 | 1.000 | n/a | **0.041** | 1.000 |
| RF | **0.045** | 1.000 | **0.002** | 1.000 | **0.013** | **0.041** | n/a | 0.480 |
| Bagged.DT.RP | 1.000 | 0.991 | 0.797 | 1.000 | 1.000 | 1.000 | 0.480 | n/a |

## 2. Μέρος Β

Στο δεύτερο μέρος της εργασία εξετάστηκε το πρόβλημα του διαφορετικού κόστους στο σύνολο δεδομένου heart .Η βιβλιοθήκη που χρησιμοποιήθηκε ήταν η **Costcla,sklearn** της python.

Το cost matrix αυτού του συνόλου δεδομένων είναι το εξής:

| | **Actual absence** | **Actual presence** |
|---|---|---|
| **Absence** | 0 | 1 |
| **Presence** | 5 | 0 |

**Πίνακας 1: Cost matrix του συνόλου δεδομένων heart**

Οι μέθοδοι που χρησιμοποιήθηκαν και αναφέρονται παρακάτω συνδυάστηκαν με τους αλγορίθμους μάθησης Random Forest, Linear SVM,Naive Bayes:

- Δίνοντας έμφαση στα παραδείγματα με το μεγαλύτερο κόστος(**CostSampling** [Oversampling, RejectionSampling], Undersampling )
- Ελαχιστοποίηση αναμενόμενου κόστους εκτιμήσεων (**ThresholdOptimization,BayesMinimumRiskClassifier)**
- Τροποποιημένη cost sensitive ταξινομητές( **CostSensitiveRandoForestClassifier**)

Τα ονόματα στην στήλη Algorithm που φαίνονται στον πίνακα [X] αποτελεσμάτων είναι οι τεχνικές που χρησιμοποιήθηκαν σε συντομογραφία πιο συγκεκριμένα παρακάτω δίνονται οι πλήρεις ονομασίες:

- **RF** : RandomForest
- **RF – O :** RandomForest Over-Sampling
- **RF – R:** RandomForest Rejection-Sampling

- **RF – U:** RandomForest Under-Sampling
- **RF - BMR**: RandomForest BayesMinimumRiskClassifier
- **RF - TO**: RandomForest ThresholdOptimization
- **RFC**: CostSensitiveRandoForestClassifier
- **LSVM**: LinearSVM
- **LSVM - O** : LinearSVM Over-Sampling
- **LSVM - R**: LinearSVM Rejection-Sampling
- **LSVM** : LinearSVM Under-Sampling
- **LSVM - BMR**: LinearSVM BayesMinimumRiskClassifier
- **LSVM - TO**: LinearSVM ThresholdOptimization
- **GNB**: NaiveBayes
- **GNB – O:** NaiveBayes Over-Sampling
- **GNB – R**: NaiveBayes Rejection-Sampling
- **GNB – U**: NaiveBayes Under-Sampling
- **GNB - BMR**: NaiveBayes BayesMinimumRiskClassifier
- **GNB - TO**:: NaiveBayes ThresholdOptimization

Το SVM τείνει να πιέζει τις προβλεπόμενες πιθανότητες μακριά από 0 και 1. Άλλα μοντέλα όπως Naive bayes έχουν την αντίθετη προκατάληψη και τείνουν να ωθούν τις προβλέψεις πιο κοντά στα 0 και 1.Αυτό βλάπτει την ποιότητα των πιθανοτήτων. Για αυτό έγινε calibration στις πιθανότητες των αλγορίθμων μάθησης Random Forest, Linear SVM,Naive Bayes με τη χρήση της βιβλιοθήκης **sklearn.calibration.**

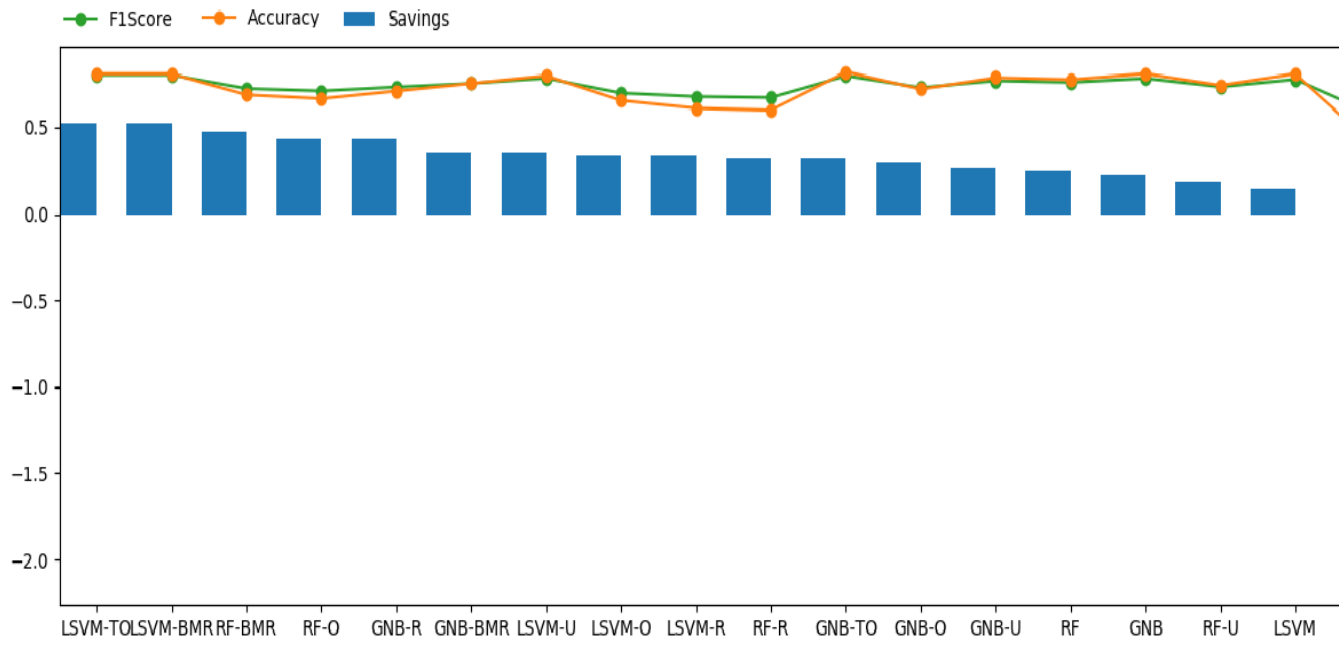
**2.1 Αποτελέσματα**

| Algorithm | F1 | Accuracy | Saving score | Cost |
|---|---|---|---|---|
| **RF** | 0.761905 | 0.777778 | 0.245283 | 40.0 |
| **RF - O** | 0.711538 | 0.666667 | 0.433962 | 30.0 |
| **RF - R** | 0.672727 | 0.600000 | 0.320755 | 36.0 |
| **RF - U** | 0.735632 | 0.744444 | 0.188679 | 43.0 |

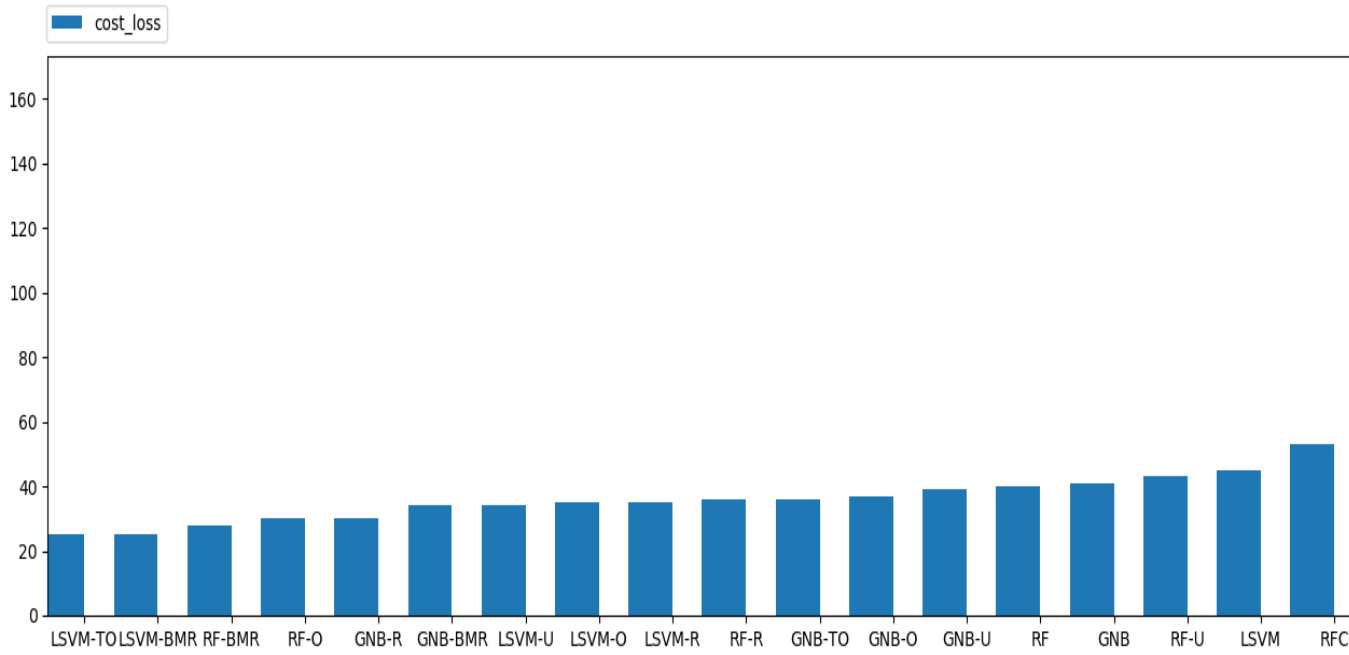
| | | | | |
|---|---|---|---|---|
| **RF - BMR** | 0.725490 | 0.688889 | 0.471698 | 28.0 |
| **RF - TO** | 0.195122 | 0.633333 | -2.113208 | 165.0 |
| **LSVM** | 0.779221 | 0.811111 | 0.150943 | 45.0 |
| **LSVM - O** | 0.699029 | 0.655556 | 0.339623 | 35.0 |
| **LSVM - R** | 0.678899 | 0.611111 | 0.339623 | 35.0 |
| **LSVM - U** | 0.785714 | 0.800000 | 0.358491 | 34.0 |
| **LSVM - BMR** | 0.804598 | 0.811111 | 0.528302 | 25.0 |
| **LSVM - TO** | 0.804598 | 0.811111 | 0.528302 | 25.0 |
| **GNB** | 0.784810 | 0.738095 | 0.226415 | 41.0 |
| **GNB - O** | 0.731183 | 0.722222 | 0.301887 | 37.0 |
| **GNB – R** | 0.734694 | 0.711111 | 0.433962 | 30.0 |
| **GNB - O** | 0.771084 | 0.788889 | 0.264151 | 39.0 |
| **GNB - BMR** | 0.755556 | 0.755556 | 0.358491 | 34.0 |
| **GNB - TO** | 0.800000 | 0.822222 | 0.320755 | 36.0 |
| **RFC** | 0.582677 | 0.411111 | 0.000000 | 53.0 |

**Πίνακας 2: κόστους**

**Πίνακας 3: Αποτελεσμάτα των μεθόδων για το πρόβλημα του διαφορετικού κόστους**




**Εικόνα 1:Διάγραμμα saving score προς f1 και accuracy από την καλύτερη απόδοση κόστους προς την χειρότερη**




**Εικόνα 2:Διάγραμμα κόστους από την καλύτερη απόδοση στη χειρότερη**

### 2.2 Συμπεράσματα

Από τις εικόνες τον αποτελεσμάτων φαίνεται πως το μοντέλο **LSVM – To** και **LVSM - BMR** καθώς έχουν το μικρότερο κόστος και έχουν πολύ καλή απόδοση F1.Αντίθετα το μοντέλο **RF-TO** δίνει τα χειρότερα αποτελέσματα καθώς έχει πολύ μεγάλο κόστος και την χαμηλότερη απόδοση F1.Γενικά, όλα τα μοντέλα ελαχιστοποιήσεις κόστους

εκτός το μοντέλο **RF-TO** της δίνουν πολύ καλή απόδοση στη μετρική του κόστους αλλά και στις μετρικής F1.

Ενδιαφέρον παρουσιάζει, ότι δίνοντας έμφαση στα παραδείγματα με το μεγαλύτερο κόστος με τις τεχνικές **Oversampling, RejectionSampling, Undersampling** καθώς χάνουν απόδοση στις μετρικές F1, Accuracy η μετρική κόστους τους μειώνεται σε σχέση με τους αρχικού αλγορίθμους RF,LSVM,GNB.

Επίσης, ο **Cost Sensitive RandoForest** ταξινομητής δεν παρουσιάζει καλά αποτελέσματα στο συγκεκριμένο πρόβλημα.

Τέλος, εκτός της μεθόδου ελαχιστοποίησης του αναμενόμενου κόστους εκτιμήσεων φαίνεται να διακρίνεται το μοντέλο LSVM.Στις άλλες μεθόδους φαίνεται τα μοντέλα LSVM,GNB να έχουν σχετικά όμοιες αποδώσεις.

## 3. Μέρος Γ

Στο τρίτο μέρος της εργασίας εξετάστηκε το πρόβλημα της ασυμμετρίας κλάσεων στο σύνολο δεδομένων creditcardfraud που περιέχει 284807 συναλλαγές με πιστωτικές κάρτες, στο οποίο μόνο το 0.172% αυτών είναι απάτες (θετική κλάση). Η ανάλυση πραγματοποιήθηκε με τη βοήθεια της βιβλιοθήκης **imblearn** της python.

Οι μέθοδοι που χρησιμοποιήθηκαν και αναφέρονται παρακάτω συνδιάστηκαν με τους αλγορίθμους μάθησης του **scikit-learn** RandomForestClassifier, LinearSVC και GaussianNB:

- **Over-sampling:** SMOTE
- **Under-sampling:** NearMiss (version=2)
- **Ensemble of samplers:** EasyEnsemble μέσω του BalancedBaggingClassifier

Στα αποτελέσματα που δίνονται παρακάτω χρησιμοποιούνται οι μετρικές precision(pre), recall(rec), specificity(spe), geometric mean(geo), και index balanced accuracy of the geometric mean(iba) για τις δύο κλάσεις.

| NearMiss-2 - RandomForestClassifier | | | | | | |
|---|---|---|---|---|---|---|
| **pre** | **rec** | **spe** | **f1** | **geo** | **iba** | **sup** |
| 0.998 | 0.008 | 0.992 | 0.016 | 0.089 | 0.007 | 71079 |
| 0.002 | 0.992 | 0.008 | 0.003 | 0.089 | 0.009 | 123 |
| 0.997 | 0.010 | 0.990 | 0.016 | 0.089 | 0.007 | 71202 |

| NearMiss-2 - LinearSVC |
|---|

| pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|
| 1.000 | 0.173 | 0.967 | 0.295 | 0.409 | 0.154 | 71079 |
| 0.002 | 0.967 | 0.173 | 0.004 | 0.409 | 0.181 | 123 |
| 0.998 | 0.175 | 0.966 | 0.295 | 0.409 | 0.154 | 71202 |

**NearMiss-2 - GaussianNB**

| pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|
| 1.000 | 0.943 | 0.813 | 0.971 | 0.876 | 0.777 | 71079 |
| 0.024 | 0.813 | 0.943 | 0.047 | 0.876 | 0.757 | 123 |
| 0.998 | 0.943 | 0.813 | 0.969 | 0.876 | 0.777 | 71202 |

**SMOTE - RandomForestClassifier**

| pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|
| 1.000 | 1.000 | 0.780 | 1.000 | 0.883 | 0.797 | 71079 |
| 0.889 | 0.780 | 1.000 | 0.831 | 0.883 | 0.763 | 123 |
| 0.999 | 0.999 | 0.781 | 0.999 | 0.883 | 0.797 | 71202 |

**SMOTE - LinearSVC**

| pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|
| 1.000 | 0.979 | 0.902 | 0.989 | 0.940 | 0.890 | 71079 |
| 0.068 | 0.902 | 0.979 | 0.126 | 0.940 | 0.876 | 123 |
| 0.998 | 0.978 | 0.903 | 0.988 | 0.940 | 0.890 | 71202 |

**SMOTE - GaussianNB**

| pre | rec | spe | f1 | geo | iba | sup |
|---|---|---|---|---|---|---|
| 1.000 | 0.975 | 0.837 | 0.987 | 0.904 | 0.828 | 71079 |
| 0.055 | 0.837 | 0.975 | 0.104 | 0.904 | 0.805 | 123 |
| 0.998 | 0.975 | 0.838 | 0.986 | 0.904 | 0.828 | 71202 |

| BalancedBaggingClassifier - RandomForestClassifier | | | | | | |
|---|---|---|---|---|---|---|
| **pre** | **rec** | **spe** | **f1** | **geo** | **iba** | **sup** |
| 1.000 | 0.987 | 0.870 | 0.993 | 0.926 | 0.868 | 71079 |
| 0.101 | 0.870 | 0.987 | 0.180 | 0.926 | 0.848 | 123 |
| 0.998 | 0.986 | 0.870 | 0.992 | 0.926 | 0.868 | 71202 |

| BalancedBaggingClassifier - LinearSVC | | | | | | |
|---|---|---|---|---|---|---|
| **pre** | **rec** | **spe** | **f1** | **geo** | **iba** | **sup** |
| 1.000 | 0.976 | 0.894 | 0.988 | 0.934 | 0.880 | 71079 |
| 0.060 | 0.894 | 0.976 | 0.112 | 0.934 | 0.865 | 123 |
| 0.998 | 0.976 | 0.894 | 0.986 | 0.934 | 0.880 | 71202 |

| BalancedBaggingClassifier - GaussianNB | | | | | | |
|---|---|---|---|---|---|---|
| **pre** | **rec** | **spe** | **f1** | **geo** | **iba** | **sup** |
| 1.000 | 0.967 | 0.846 | 0.983 | 0.904 | 0.828 | 71079 |
| 0.043 | 0.846 | 0.967 | 0.082 | 0.904 | 0.808 | 123 |
| 0.998 | 0.967 | 0.846 | 0.982 | 0.904 | 0.828 | 71202 |

Τα συμπεράσματα που μπορεί να εξαχθούν είναι τα εξής:

- Η χρήση της μεθόδου NearMiss παρουσιάζει γενικά φτωχά αποτελέσματα, με εξαίρεση όταν γίνεται χρήση Naïve Bayes που δίνει geometric mean > 0.85.
- Η χρήση της μεθόδου SMOTE αν και αυξάνει αρκετά τον χρόνο εκπαίδευσης δίνει καλά αποτελέσματα με σχετικά μικρή διακύμανση και geometric mean > 0.85 για όλους τους ταξινομητές. Επίσης παρουσιάζει τα καλύτερα score για τις μετρικές geometric mean και index balanced accuracy όταν εφαρμόζεται ο LinearSVC.
- Η χρήση της μεθόδου EasyEnsemble είναι πολύ γρήγορη σε χρόνο εκπαίδευσης και δίνει καλά αποτελέσματα με μικρή διακύμανση. Οι τιμές της μετρικής geometric mean είναι μεγαλύτερες του 0.9, ενώ και αυτές για την μετρική index balanced accuracy είναι αρκετά υψηλές για τις δύο κλάσεις.