Data Mining - Classification

Mounir Madmar (20190457)

April 2022

1 Data Preprocessing

All the missing values where replaced with the most frequent values for that particular feature (column) and all the category features such as workclass or race where labeled using a label encoder.

2 Scoring

The data is very imbalanced around 75% of the training data is low income and 25% high income. This is why accuracy is not a good metric to rank the classifiers. Instead the estimated profit of each classifier is used to rank them using the True Positives and False Positives from the confusion matrix.

3 Classifiers

In this section I will discuss the classifiers that were used for this project. All these classifiers were tested using the GridSearchCV object which performs a grid search together with a 5-fold cross-validation to evaluate and find the best parameters for the given classifier.

3.1 KNeighbors Classifier

First the KNeighbors Classifier is used and different number of neighbors are tested as parameter. Using only 5 neighbors gets the best results.

3.2 Decision Tree Classifier

I wanted to find the optimal splitting criterion for this data using the Decision Tree Classifier. The gini entropy yields the best outcomes in this scenario.

3.3 Categorical Naive Bayes

Since most of the features are categories I wanted to try this classifier and see how it performs. For this I did a 5-fold cross validation with the default parameters and saved the average metrics.

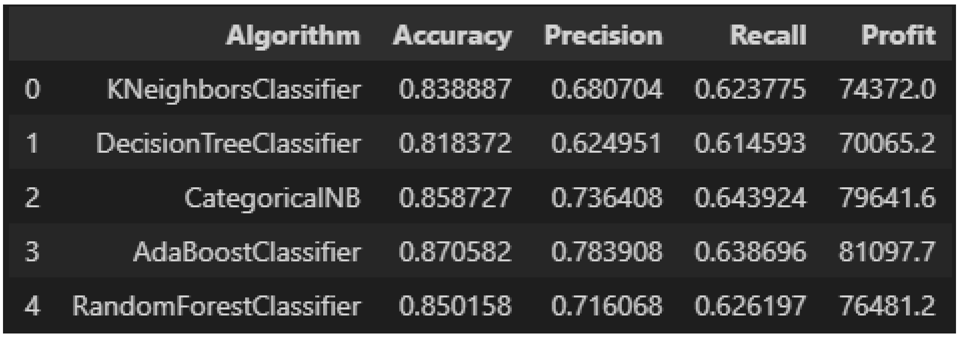
3.4 AdaBoost Classifier

I wanted to boost the Decision Tree classifier using the AdaBoost Classifier. For this 50, 100 and 1000 estimators are tested. As I was expecting a higher number of estimators gave a better result. I kept it to max 1000 because the computing times increase a lot if you use more estimators.

3.5 RandomForest Classifier

For this last classifier I wanted to see if we could get better results using multiple Decision Trees. Random Forest is the perfect algorithm for this, I tested three different number of trees 100, 200 and 1000. Again as I was expecting higher number of estimators gave better results but with the cost of higher computation times.

4 Metrics



	Algorithm	Accuracy	Precision	Recall	Profit
0	KNeighborsClassifier	0.838887	0.680704	0.623775	74372.0
1	DecisionTreeClassifier	0.818372	0.624951	0.614593	70065.2
2	CategoricalNB	0.858727	0.736408	0.643924	79641.6
3	AdaBoostClassifier	0.870582	0.783908	0.638696	81097.7
4	RandomForestClassifier	0.850158	0.716068	0.626197	76481.2

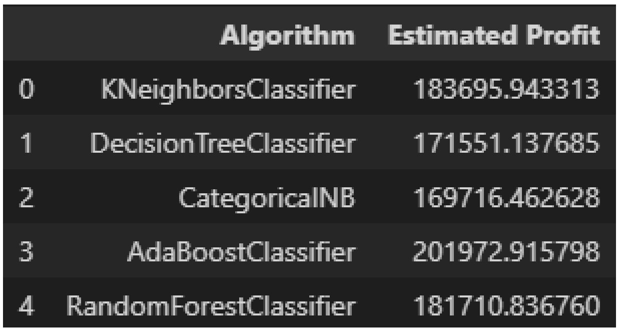
Figure 1: Average metrics of each classifier during cross-validation

Figure 1 shows the average metrics of each classifier with the best parameters during the cross-validation. In this case AdaBoost with 100 estimators performs the best since it achieves the highest profit. Note that this profit is profit on 20% of the dataset because we use a 5-fold cross-validation so each test fold is 20% of the dataset.

5 Potential Customers

Every model was then used to predict the labels of the potential customers. The precision obtained during the evaluation/optimization was used to calculate the estimated profit. For example if a model has 70% precision and labels 1000 customers as high income then we will estimate that of those 1000 customers that we send the promotion 700 (True Positives) of them are actually high income and 300 (False Positives) low income and then calculate the profit based on that.

6 Metrics



	Algorithm	Estimated Profit
0	KNeighborsClassifier	183695.943313
1	DecisionTreeClassifier	171551.137685
2	CategoricalNB	169716.462628
3	AdaBoostClassifier	201972.915798
4	RandomForestClassifier	181710.836760

Figure 2: Estimated profit of each classifier on the potential customers

Figure 2 shows the estimated profit on the potential customer for each classifier. Again as during the evaluation/optimization AdaBoost yields the highest estimated profit. AdaBoost (with 1000 estimators) is used to create the txt file with all the row ids of the customers that we send the promotion to. The single estimate that is asked in the assignment is thus 201973EUR.

7 Implementation

All this was implemented using a Jupyter Notebook which is properly documented. This can be found in the following public github repo: https://github.com/MrMounir19/dm-classification