Data Mining Assignment 1: Classification

Viktor Hura - 20191842

April 16th, 2023

Introduction

In this report I will detail my progress and solution for the classification assignment.
This report will go over the context of the assignment, analysis of the provided data, pre-processing steps taken, model comparison pipeline, and finally evaluation of the chosen model followed by my solution and final performance estimate.
The solution was written in python using the scikit-learn library, the code for which is referenced at the end of this report.

Contents

0 Exploratory Data Analysis **1**

1 Pre-processing **2**

2 Model Comparison **2**
2.1 Models 2
2.2 Evaluation Metric 2
2.3 Pipeline 2
2.4 Results 3

3 Final Model & Evaluation **3**

Estimate & Conclusion **3**

Code **4**

Appendix **4**

Context

For this assignment, two datasets were provided, *existing customers* and *potential customers*. These contain the demographic information of 32562 and 16282 individuals respectively. The datasets contain the same columns, except for the potential customers missing the "income" column, which refers to the class of income of an individual, either "<=50K" (lower) or ">50K" (high).

Using existing customers as training data, we wish to classify the income class of the potential customers and send promotional packages to them.
The following details were provided:

Sending a promotional package costs 10 euro.
A high income individual tends to accept the offer 10% of the time, where as a lower income individual will only accept 5% of the time.
If a high income individual accepts the offer, we will gain on average 980 euro. If a lower income individual accepts the offer, we will on average lose 310 euro.

Given these details, I have made the following cost matrix:

Cost	Predicted class		
Actual class		high	low
	high	-88	0
	low	25.5	0

Figure 1: Cost matrix

For false negatives and true negatives, the cost is 0 as you would not send a promotion if someone was classified as lower income, thus no gain or loss.
For true and false positives I calculated the cost using the following formula:

$$C = P(\text{accept offer}|\text{class}) * (\text{average loss for class}|\text{offer accepted}) + (\text{cost of promotional package})$$

$$C(\text{True Positive}) = 0.10 * -980 + 10 = -88$$

$$C(\text{False Positive}) = 0.05 * 310 + 10 = 25.5$$

0 Exploratory Data Analysis

There training data is imbalanced, about 24% of the rows are high income. It is something that must be taken into account.

The existing customers and potential customers have very similar distributions of their features. Which gives credence that a model performing well on the training dataset, would perform well on our potential customers.
Both datasets also have missing values in the *workclass*,*occupation* and *native-country* columns. The native-country column is not very descriptive as 91% of rows are labelled "United-States". So the whole column can be discarded. In the remaining two columns I will impute the missing values.

Finally, there seems to be unusual outlier values of "99999" in the capital-gain columns of both datasets, which can be seen on this boxplot(2).

I assume this to be a clerical error and I will replace all occurrences of this value with the median of the column.

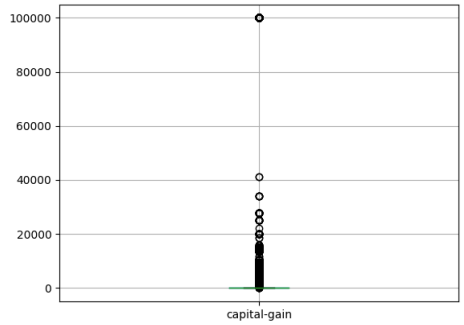

Figure 2: capital-gain values

1 Pre-processing

In this section I will describe the pre-processing steps taken for both datasets.

First I replaced the unusual capital-gain values by the median of their column.
I dropped the native-country column as mentioned because I don't believe there is enough diversity there to be meaningful for the classifier. The *education* column was dropped as well because it is made redundant by *education-num*.
Missing values of the workclass and occupation columns were imputed using the "missForest" algorithm.

Numerical columns age, education-num, capital-gain, capital-loss and hours-per-week were normalised by their respective largest values in the training dataset. All remaining categorical columns were one-hot encoded as I don't believe there is any ordinality in those features.

Finally the class column is encoded as "True" for high income and "False" for lower income.

Using the Kendall correlation coefficient, I analysed which features have a monotonic relationship with income class:

Feature	Kendall coefficient >0.2
age	0.225235
education-num	0.291662
capital-gain	0.247726
hours-per-week	0.238273
marital-status_Married-civ-spouse	0.444696
marital-status_Never-married	-0.318440
relationship_Husband	0.401035
relationship_Own-child	-0.228532
sex_Female	-0.215980
sex_Male	0.215980
occupation_Exec-managerial	0.21324

Figure 3: Feature correlations

This provides us an idea of which features might be important for the models to consider and can be useful to compare during feature selection. It suggests that of the 46 features in the pre-processed dataset, only a fraction might be important for predicting class.

2 Model Comparison

2.1 Models

The following classification models were considered for this assignment:

- Dummy Classifier (as a baseline)
- KNN
- Decision Tree
- Random Forest
- Histogram Gradient Boost
- Extreme Boost
- Linear Discriminant Analysis
- Logistic Regression
- Linear SVM
- SVM

2.2 Evaluation Metric

As the goal of this assignment is to maximise revenue, I have chosen a custom performance metric by which I will be assessing these models:

$$\text{cost-extracted-score} = \frac{C(TP) * TP + C(FP) * FP}{C(TP) * (TP + FN)}$$

Where C(TP) = -88, C(FP) = 25.5 as determined in Figure 1.

This metric represents the ratio between the profit generated using the model predictions and the theoretical maximum profit that can be generated on the set,
if you were to perfectly predict each high income client.

For maximising profit, we wish this metric to approach as close as possible to 1.

2.3 Pipeline

The pipeline for testing a model looks as follows:

First the existing customers data is split into training and test sets where 75% is training and the remainder is test. This is done in a stratified manner to account for the class imbalance, such that in each

set roughly 24% of the rows are high income.

From a predefined grid, all combinations of hyper-parameters are generated for this model.
For each hyper-parameter combination (optionally*) relevant features are selected using forward feature selection.
After which the training dataset with the retained features, is split again using stratified k-fold.
With k=10, it is split in training and evaluation sets to evaluate the current hyper-parameter combination based on the average cost-extracted-score metric 2.2, after which the model is discarded.

After all hyper-parameter combinations have been evaluated, the set of features and hyper-parameters with the highest cost-extracted-score will be selected.

A new model will then be trained on the whole training set using these hyper-parameters and evaluated on the test set that was set aside earlier and had not been used used during training, hyper-parameter optimisation nor feature selection.

*Note that for ensemble methods Random Forest, Histogram Gradient Boost and Extreme Boost, no forward feature selection was performed as these models have built in feature selection mechanisms.

2.4 Results

The test results can be found in the Appendix 3, though they may differ slightly depending on the exact seeds used when running the code.

The ensemble methods performed best, followed by logistic regression and linear SVM.
Histogram Gradient Boost and XGBoost tie for first place which makes sense.
I chose the Histogram Gradient Boost as it showed less signs of over-fitting, is easier to optimise and was slightly faster to train.

3 Final Model & Evaluation

Having chosen the Histogram Gradient Boost model, I split the existing customers once more into 75% training and remainder test in a stratified manner.
After fitting a fresh model on the training set, these are the results from evaluating it on the test set:

Histogram Gradient Boost	
extracted-value -score	0.7063
precision	0.5981
recall	0.8770
F1	0.7112
ROC AUC	0.8451
accuracy	0.8285
EVPC	-42.3867

Figure 4: Histogram Gradient Boost Model Results

To be able to make a reasonable estimate of the profit that the model will generate, I created the so called "EVPC" score, which stands for Expected Value per Positive Classification.
This score is calculated as a linear interpolation between the cost of a true positive and cost of a false positive, using the evaluated precision:

$$\text{EVPC} = (\text{C(TP)} * \text{precision}) + (\text{C(FP)} * (1 - \text{precision}))$$

To estimate the expected profit for the model, I will count the amount of positive predictions that the model made and multiply it with the EVPC score.

Finally, using permutation feature importance method, I investigated what features are considere important to this model.
As seen on graph 6, only a few features are of importance for predicting the income class, in the following order:

- Married civ spouse
- Capital gain
- Education num
- Age
- Capital loss
- Hours per week
- Occupation farming fishing
- Occupation exec-managerial

It is important to see that this model is not biased against race or sex.

Estimate & Conclusion

Applying the fitted model on the potential customers, the model made 5599 positive predictions.
Thus based on an EVPC score(3) of 42.3867, my estimated profit generated with this model is **237323 euro**.

Code

All the code and data is available in the following repository.

Appendix

	cost-extracted -score (validation set)	**cost-extracted -score (test set)**	recall	precision	F1	ROC AUC	accuracy	best hyper-params	feature selection
Dummy	0.0865	0.0862	1.0	0.2407	0.3880	0.5	0.2407	{'constant': True, 'strategy': 'constant'}	['age']
KNN	0.2058	0.5118	0.5949	0.6747	0.6323	0.7519	0.8334	{'n_neighbors': 16, 'weights': 'distance'}	['marital-status_Divorced', 'relationship_Husband', 'relationship_Unmarried', 'relationship_Wife', 'race_White']
Decision Tree	0.5335	0.5168	0.6230	0.6297	0.6263	0.7533	0.8210	{'max_depth':9}	['education-num', 'capital-gain', 'capital-loss', 'marital-status_Married-civ-spouse', 'occupation_Exec-managerial']
Random Forest	0.6770	0.6765	0.8939	0.5437	0.6761	0.8280	0.7938	{'class_weight': 'balanced', 'max_depth': 10, 'max_features': 10, 'max_samples': 0.1, 'n_estimators': 2000}	N/A
Hist Grad Boost	0.6982	0.7003	0.8719	0.5954	0.7076	0.8420	0.8265	{'categorical_features': [...], 'class_weight': 'balanced', 'max_depth': 4, 'max_iter': 500}	N/A
XGBoost	0.7045	0.7001	0.8602	0.6088	0.7130	0.8424	0.8333	{'colsample_bytree': 0.1064, 'n_estimators': 100, 'scale_pos_weight': 3.15, 'subsample': 1.0, 'tree_method': 'hist'}	N/A
LDA	0.5115	0.5354	0.6015	0.7251	0.6575	0.7646	0.8491	{'shrinkage': 0.45, 'solver': 'lsqr'}	['education-num', 'capital-gain', 'capital-loss', 'marital-status_Married-civ-spouse', 'occupation_Exec-managerial']
Logistic Regression	0.5115	0.6561	0.8526	0.5570	0.6737	0.8187	0.8012	{'C': 0.3, 'class_weight': 'balanced', 'max_iter': 1000, 'penalty': 'l2', 'solver': 'liblinear'}	['age', 'education-num', 'capital-gain', 'hours-per-week', 'marital-status_Married-civ-spouse']
Linear SVM	0.6166	0.6581	0.8592	0.5532	0.6730	0.8195	0.7990	{'C': 0.4, 'class_weight': 'balanced', 'dual': False, 'fit_intercept': True}	['education-num', 'capital-gain', 'marital-status_Married-civ-spouse']
SVM	0.4834	0.5023	0.5648	0.7235	0.6343	0.7481	0.8432	{'kernel': 'rbf'}	['capital-gain', 'capital-loss']

Figure 5: Model comparison results

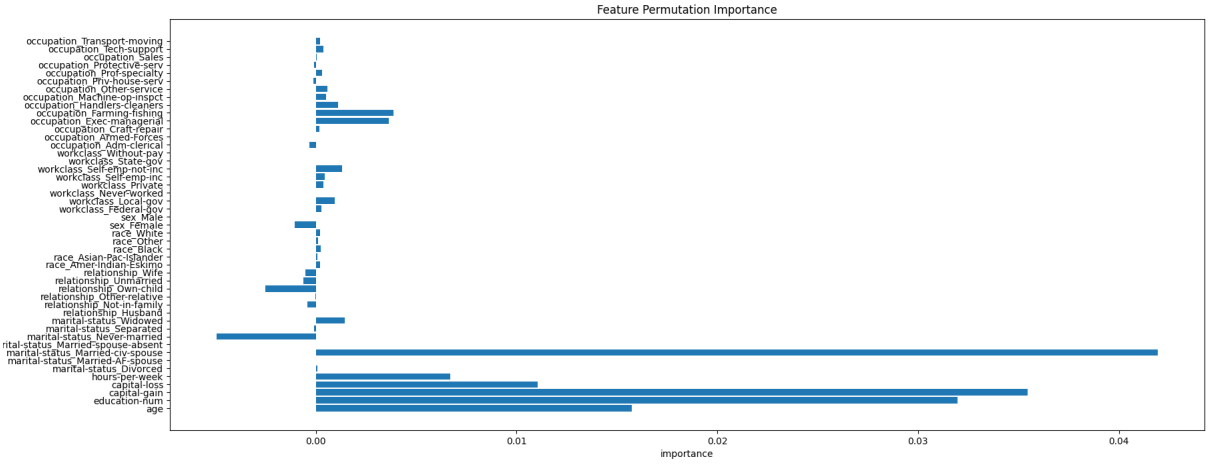

Figure 6: Feature Permutation Importance for the final model