Mathematics in Machine Learning

Adult dataset analysis

Marco Di Nepi
277959

September 2020

Contents

1 Introduction

The goal of this project it to analyze *Adult*, a well known dataset created by Barry Becker (Data Mining and Visualization, Silicon Graphics) using data from the 1994 Census database. Through machine learning algorithms and supervised techniques, we will try to predict whether a person makes more or less than 50K dollars in one year, that is a binary classification problem. We will go through an entire data science pipeline, from data exploration up to the choice of the best supervised machine learning algorithm for this task.

2 Data Exploration

The dataset is composed by 48842 samples, each of them associated with one specific individual. The 15 features are the following:

1. Categorical attributes
    - **Workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
    - **Education**: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
    - **Marital-status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
    - **Occupation**: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
    - **Relationship**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
    - **Sex**: Female, Male
    - **Native-country**: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
    - **Race**: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
2. Numerical attributes
    - **Age**
    - **Capital-gain**
    - **Capital-loss**
    - **Fnlwgt**: The weights on the CPS files

- **Hours-per-week**: Individual's working hour per week

Moreover, 2089 rows contain at least one unknown value for native country, workclass or occupations: these line have been dropped.

	age	fnlwgt	education_num	capital-gain	capital-loss	hours-per-week
count	42996.000000	4.299600e+04	42996.000000	42996.000000	42996.000000	42996.000000
mean	38.557563	1.897924e+05	10.119662	1103.576216	89.429598	40.930203
std	13.205026	1.057083e+05	2.552828	7523.598395	406.858585	11.993394
min	17.000000	1.349200e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.173675e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.782465e+05	10.000000	0.000000	0.000000	40.000000
75%	47.000000	2.380250e+05	13.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

Figure 1: Statistics of the numerical attributes

Since many algorithms are sensitive to unbalanced data, the distribution of the classes is plotted. It can be noticed that more than 75% of people earn more than 50k and, as a consequence, we'll need some rebalancing method (undersampling or oversampling).

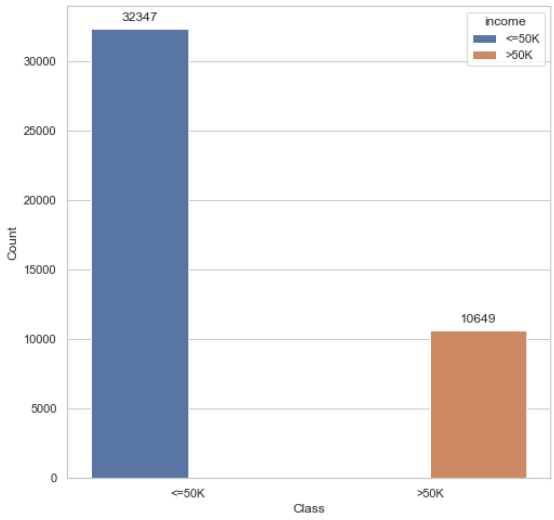
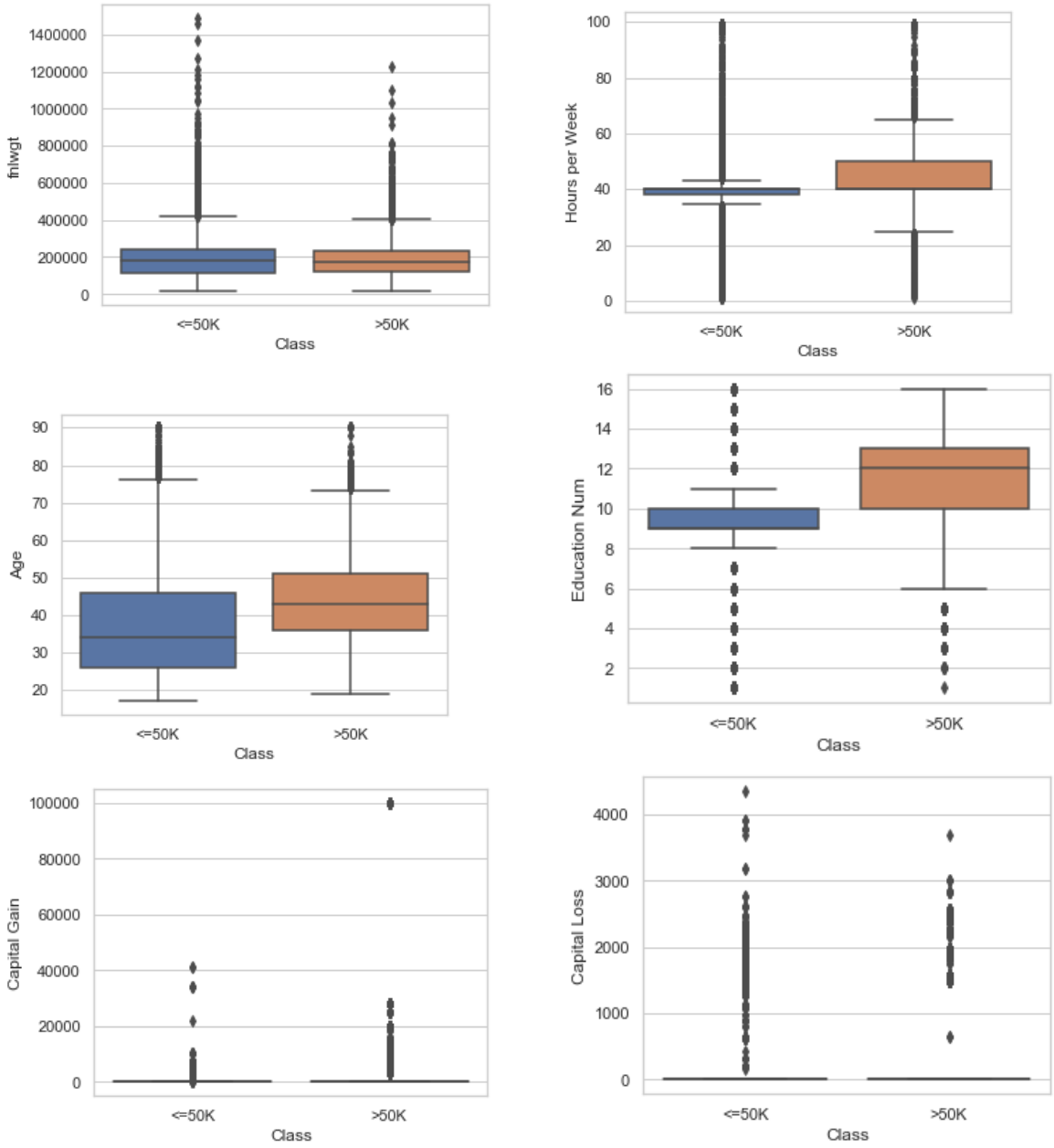
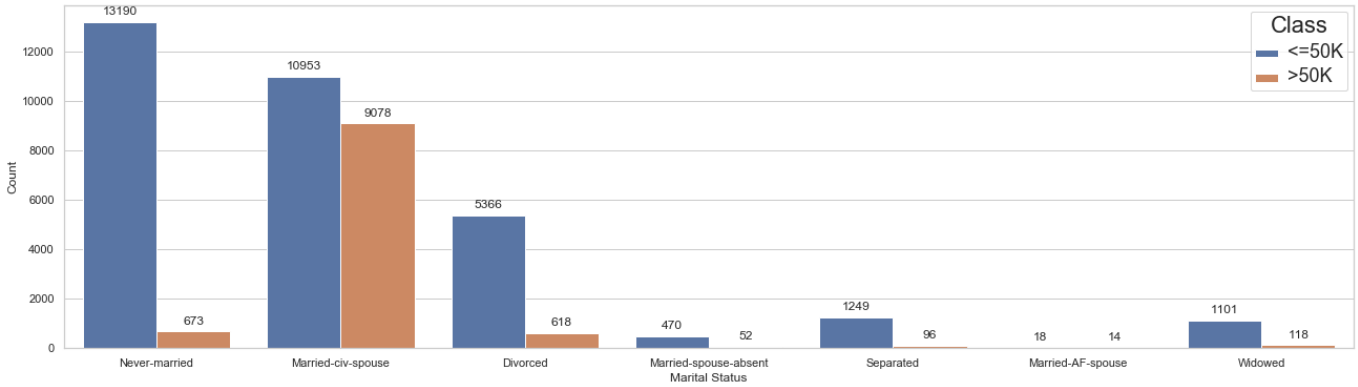


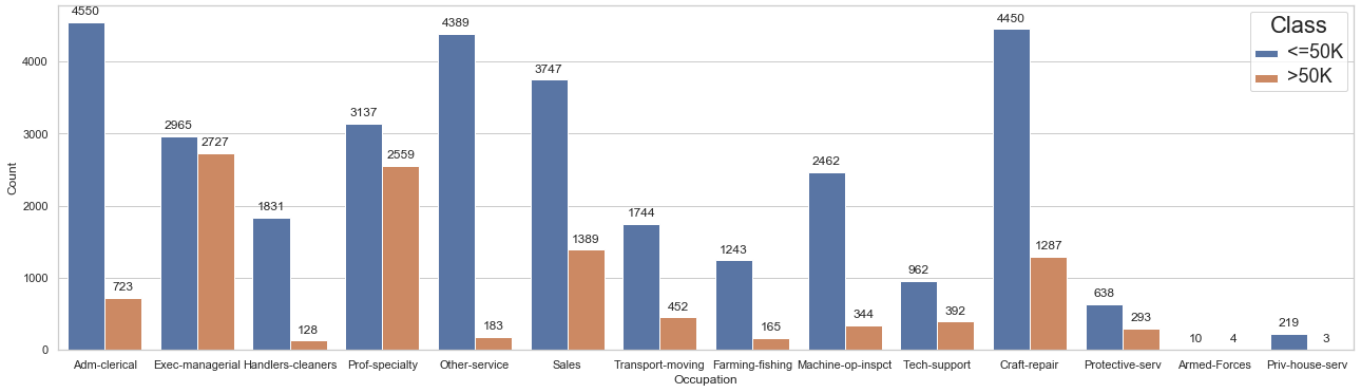
Figure 2: Number of samples for each class

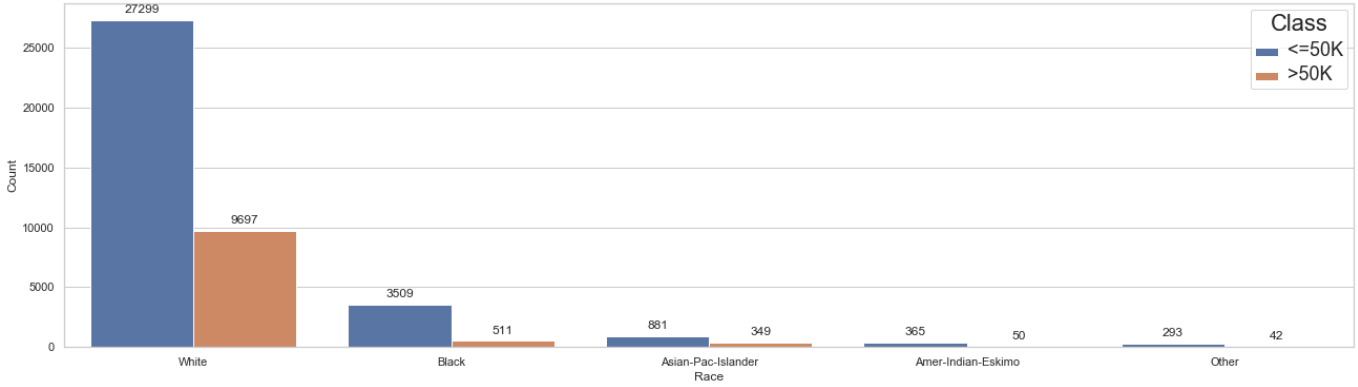
The distributions of all the attributes have been plotted as well: for the numerical features boxplots have been used, while for the categorical features we have occurrences count.

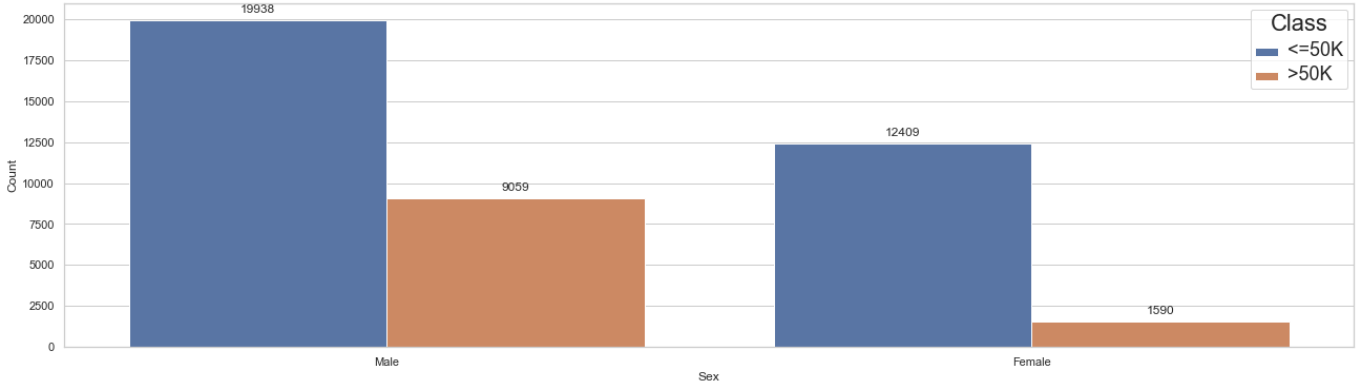


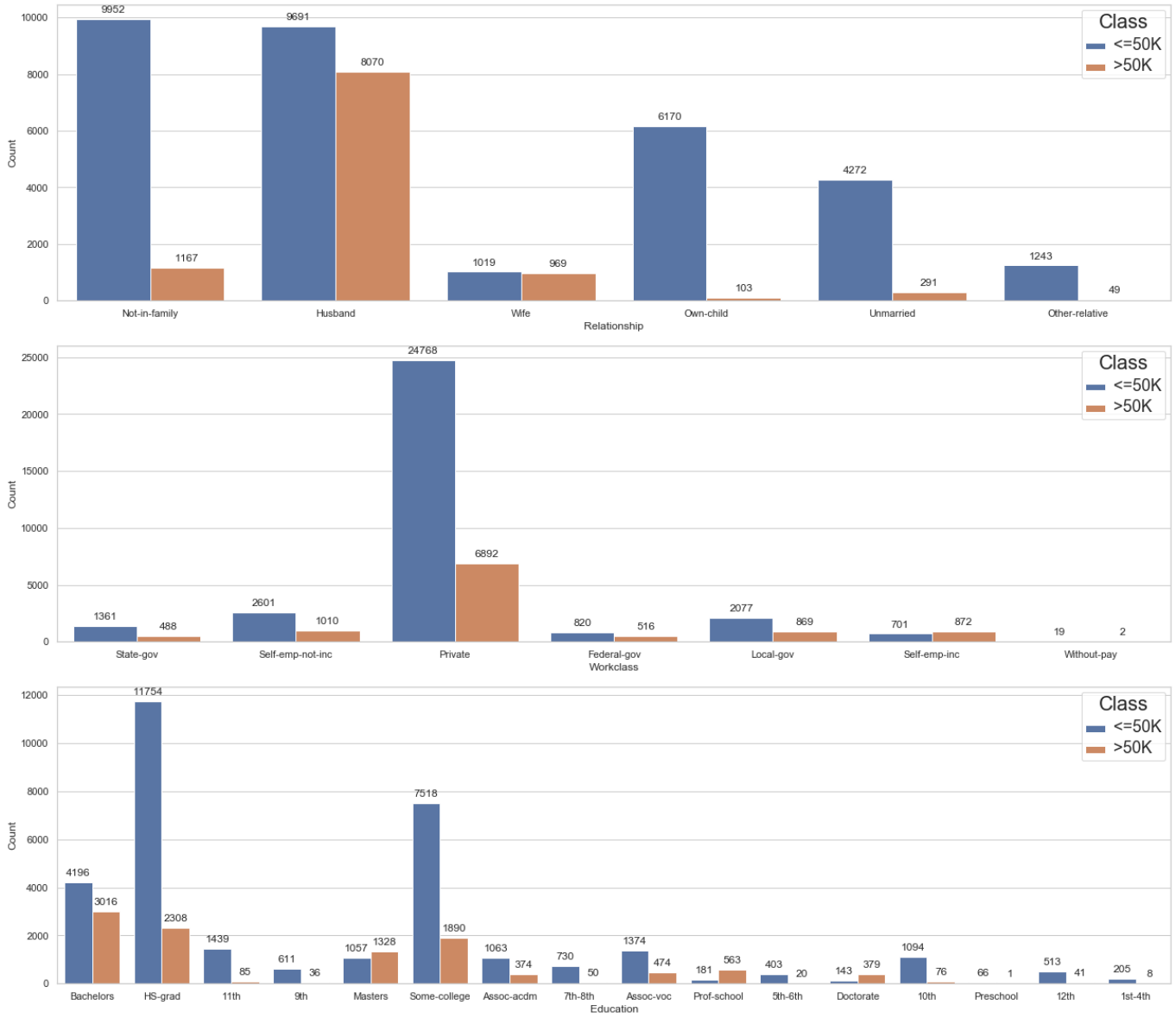
It can be noticed, for example, that there is a significant difference in the mean ages and hours-per-week of income group greater than 50K and income group smaller than 50K, while the boxplots capital loss, capital gain and fnwgt are very similar.


Class
<=50K
>50K
Count
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed
Marital Status
13190
673
10953
9078
5366
618
470
52
1249
96
18
14
1101
118


Class
<=50K
>50K
Count
Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
Craft-repair
Protective-serv
Armed-Forces
Priv-house-serv
Occupation
4550
723
2965
2727
1831
128
3137
2559
4389
183
3747
1389
1744
452
1243
165
2462
344
962
392
4450
1287
638
293
10
4
219
3


Class
<=50K
>50K
Count
White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other
Race
27299
9697
3509
511
881
349
365
50
293
42


Class
<=50K
>50K
Count
Male
Female
Sex
19938
9059
12409
1590



Regarding the categorical attributes, it's clear, as instance, that higher education means higher income (see master, prof-school, doctorate) and that for females earning more than 50K is more difficult.

2.1 Feature Correlation

A very useful way to analyze the correlation between numerical features is the correlation matrix. For each pair of attributes we can compute the covariance and divide it by the product of the standard deviations.

$$Cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \qquad \rho = \frac{Cov(x,y)}{S_x S_y}$$

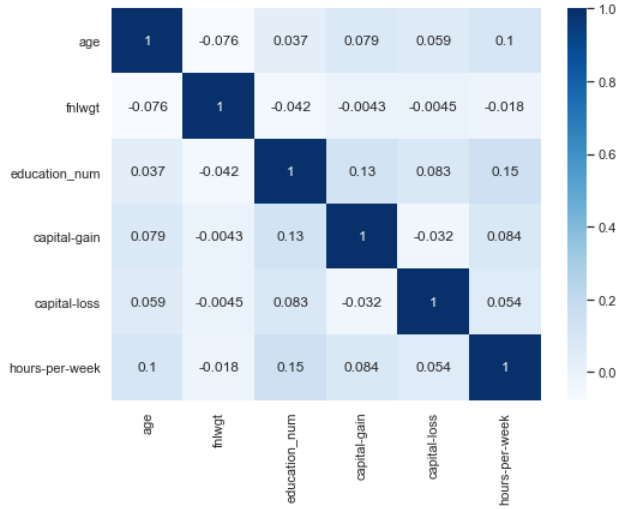
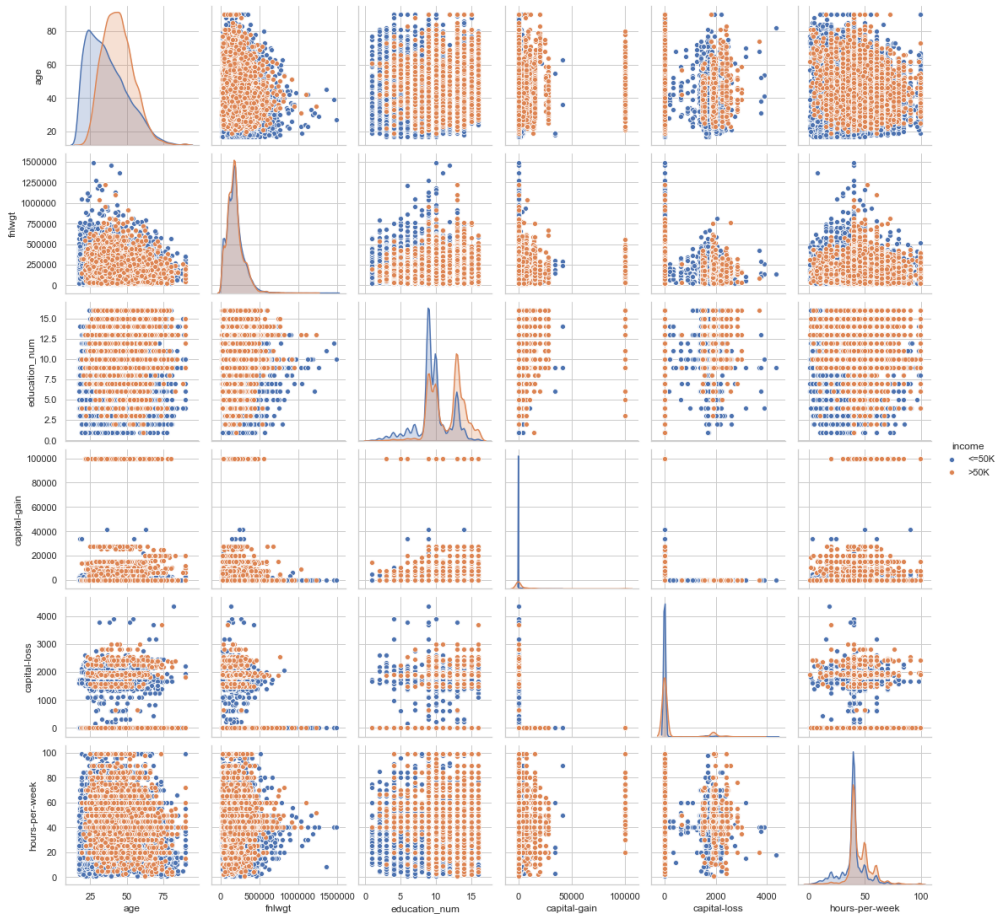


Figure 3: Correlation Matrix

However, we are not able to extract useful informations from it: All the numbers are near to zero and there are not strong positive or negative correlations.

Finally, to understand the relations between the attributes we can use a pairplot.



3 Data Preparation

3.1 Standardization

A good practise before applying machine learning algorithms is to perform a scaling of the numerical features. In this way they will have the same range of values. A simple method is a min-max normalization:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

3.2 Principal Component Analysis

PCA is an unsupervised technique commonly used for dimensionality reduction. The objective of this procedure is to find a data representation in a lower dimensional space. Since we want to lose as little information as possible, we project the data to n vectors such that the first vector (first principal component) has the direction of the largest variance of the data, while each subsequent principal vector is orthogonal to the previous one and goes in the direction of largest variance of the residual subspace. As a consequence, the principal components form an orthonormal basis in which the dimensions are uncorrelated.

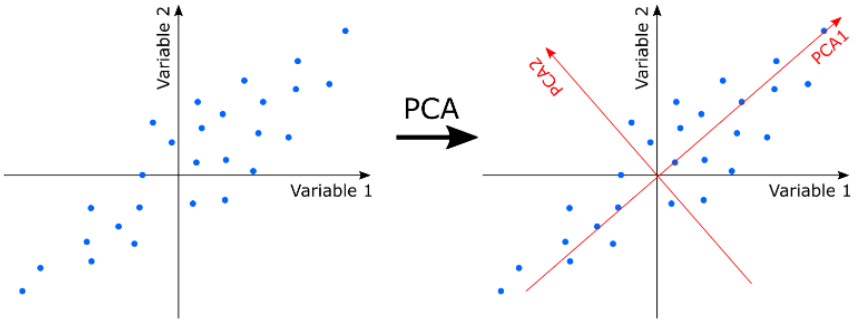
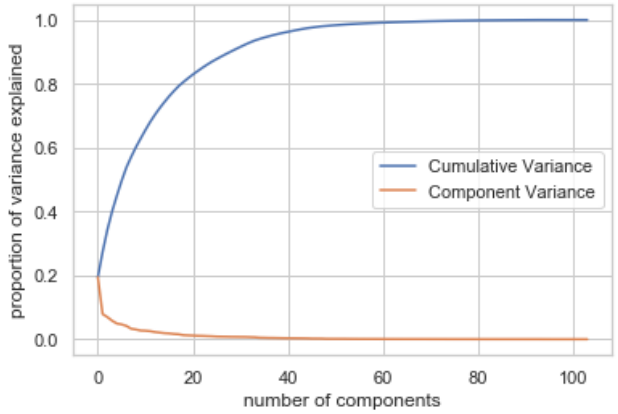


Figure 4: Example of PCA in 2 dimensions

In practise the pca algorithm is composed by the following steps:

1. compute the sample mean $\mu = \frac{1}{n} \sum x_i$
2. center the data $z_i = x_i - \mu$
3. Compute the scatter matrix $S = \sum_{i=1}^{n} z_i z_i^t$
4. Compute the eigenvectors corresponding to the largest k eigenvalues of S: $E = [e_1, e_2 ... e_k]$
5. The final result is $y = E^t z$



The graph above shows the proportion of variance explained by each component. With 40 components more than 90% of variance is explained.

3.3 Outliers management

Another important step of data processing is outliers removal. Outliers are extreme data that deviate from the other observations. Outliers can be generated by novelties in the data, but also by experimental errors or human mistakes. There are many methods that are able to detect outliers in different ways. Z-score indicates how many standard deviations a point is from the sample means. If the value is bigger then a certain threshold is discarded. Interquartile range consist in computing the difference between the first and the third quantile. Outliers here are defined as observations that fall below Q1 1.5 IQR or above Q3 + 1.5 IQR. In a boxplot they are indicated as extreme individual points.

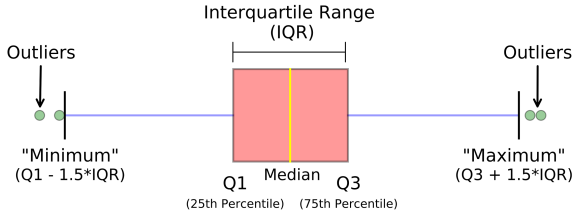


Figure 5: Parts of a boxplot

A more advanced solution is isolation forest. The main idea behind this algorithm is that outliers should be easier to separate from the rest of the data. It builds a set of isolation trees (iTrees) that split the sample in partitions recursively according to an attributed and a threshold chosen randomly. The anomalous points are the ones with a smaller path length from the root of the tree. More formally, we can compute for each point the anomaly score and mark it as an outlier if the quantity is very close to 1.

$$s(x, m) = 2^{\frac{-E(h(x))}{c(m)}}$$

3.4 Synthetic Minority Oversampling Technique

We have seen that the dataset is strongly unbalanced towards the class $\leq 50K$. In order to avoid overfitting and wrong prediction, which are often taken considering the majority class, the data must be rebalanced. It is important to remember that rebalancing should be applied only on the training data while the test will stay untouched. Regarding oversampling, a popular method is SMOTE: starting from a minority class sample, the algorithm create a new sample as a combination of the first point and one of his k neighbors on the high-dimensional lines connecting the data points: $x_{new} = x_i + \lambda(x_{zi} - x_i)$ with $\lambda \in [0, 1]$

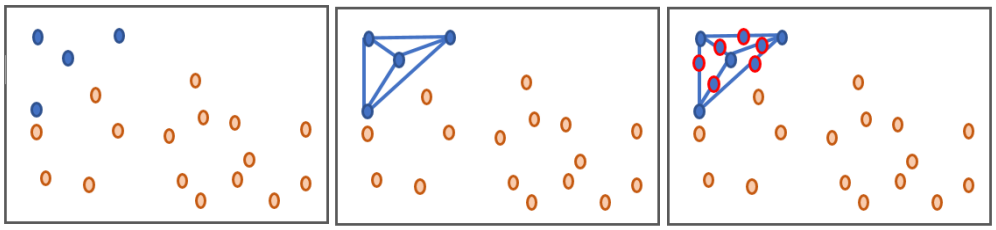


Figure 6: Oversampling with SMOTE

3.5 Cluster Centroids

Cluster centroids is an undersampling technique based on computing for the majority class the centroid, obtained as the average of the points in the feature space. Therefore, the points that are far from the centroid are considered less important and are removed up until the sample is balanced.

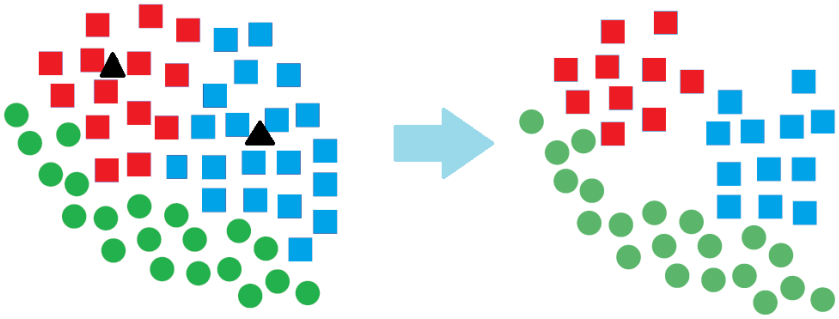


Figure 7: Undersampling with Clustere Centroids

4 Methodology

4.1 Metrics

A metric is a functions that measure how well a method performs. One of the most used is the accuracy score, that is just the ratio between the number of correctly classified samples and the total number of samples. The main problem of accuracy is that it does not work well with unbalanced dataset.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

TP: True Positive TN: True Negative
FP: False Positive FN: False Negative
A possible alternative is F1, that is worth using since it takes into account both recall and precision.

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

4.2 Cross Validation

In order to evaluate the different models and tune their parameters, k-fold with stratify has been used. The entire dataset is divided in k parts and at each iteration one subsample is used as validation set while the remaining parts are used as training set. This process is repeated k times and the final score is obtained by taking the mean of the k results. This solution has less overlap between the different sets with respect to Leave-one-out, so it has lower variance and give a good estimate of the performances. Since we want training and test to have similar distributions, we use stratify to maintain the proportions of the data, considering the binary classification.

Figure 8: K-fold Cross Validation with k = 10

5 Analysys of classification models

5.1 Decision Tree

Decision tree is a simple surpervised algorithm that models a set of sequential and hierarchical decision rules: This method divides the predictor space in non-overlapping regions that are high dimensional rectangles. The label assigned is the most commonly occurring class in that region. The algorithm uses a greedy approach and at each step chooses the best predictor to split according to some criteria. The most commons one are:

1. Gini Index: $\sum_{k=1}^{K} p_{mk}(1 - p_{mk})$ which represent a measure of purity of the node
2. Cross Entropy: $-\sum_{k=1}^{K} p_{mk} log(p_{mk})$

This method is simple and interpretable but tends to overfits and the performances are generally poorer than the ones obtained with more advanced algorithms.

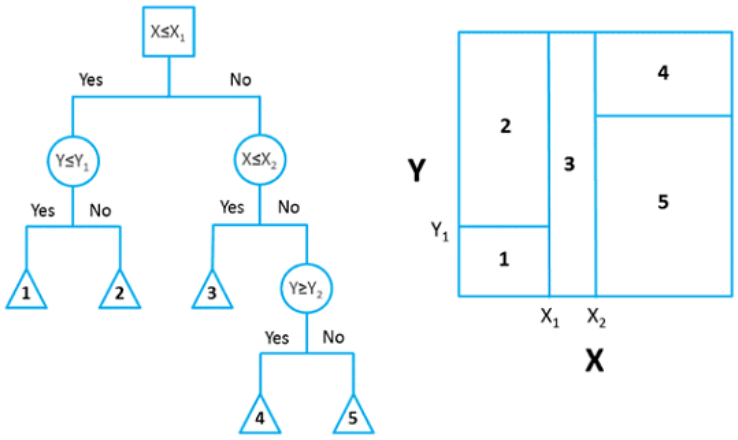


Figure 9: Basic scheme of a decision tree in a bidimensional data space

In addition to the criterion, another important hyperparameter is the maximum depth of the tree. Since we want to build a model the is able to generalize, we cannot make the tree grow too much to avoid overfitting. As said before, the hyperparameters tuning is made using k-fold cross validation with stratify while the final model is tested on a separate set that contains 30% of the initial dataset. The best performing combination found is criterion=gini and max-depth=10.

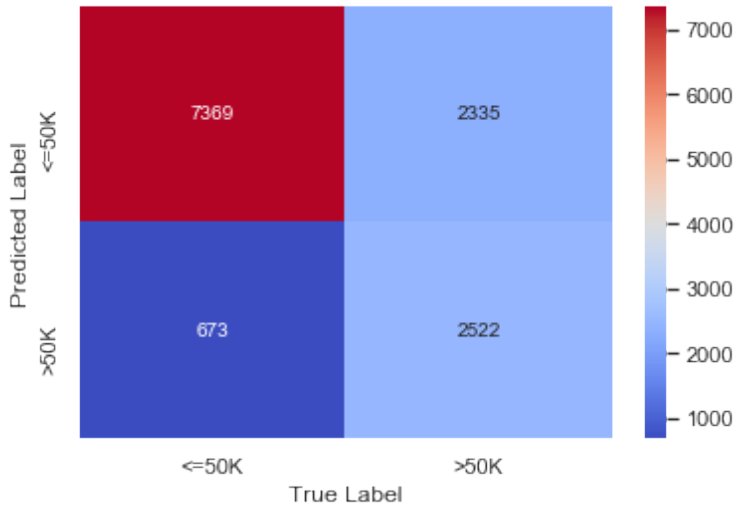


Figure 10: Decision Tree confusion matrix. F1 Score: 0.62

5.2 Random Forest

A popular way to overcome the limitation of decision trees, is building a random forest. Exploiting the bagging technique, a certain number of trees are created, and each of them is trained on a different bootstrapped set of the training data. More than that, each time a split is considered, only m random predictors are taken into account (typically $\sqrt{p}$), this allow to build uncorrelated trees and as a consequence a more robust model with smaller variance. The prediction is made aggregating the decisions of the different trees by majority voting. The hyperparameters chosen are the same as before and the number of estimators has been set to 300. As showed by the confusion matrix below, the F1 score is slightly better that the one obtained with decision tree.

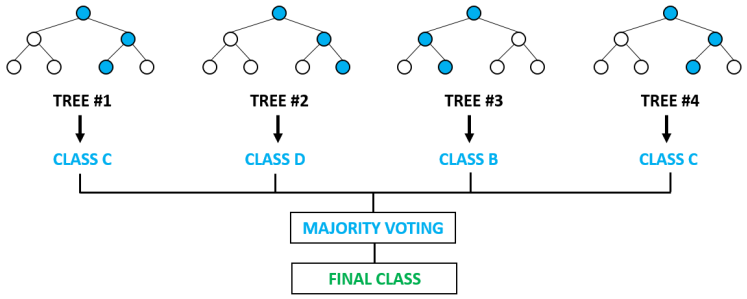


Figure 11: Random Forest example

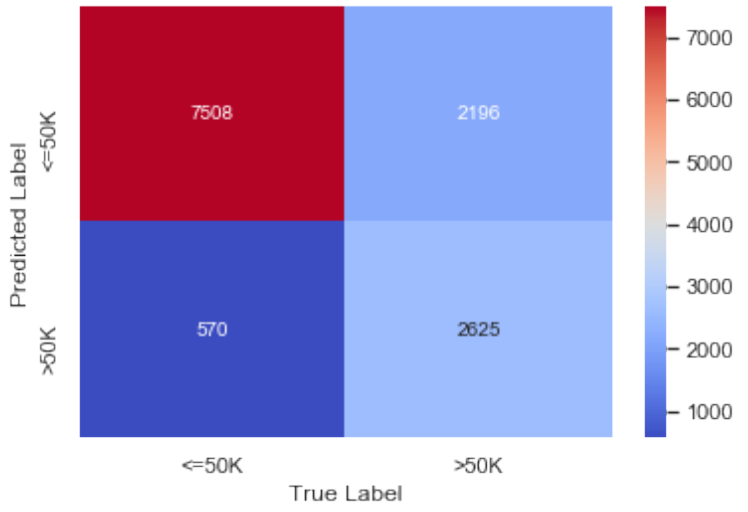


Figure 12: Random Forest confusion matrix. F1 Score: 0.65

5.3 Logistic Regression

Logistic regression is a regression model used for binary classification based on probability. Differently from the linear regression, we want the result to be between 0 and 1, so we cannot use just a linear function but instead, we use the logistic function (sigmoid) which takes any real value and maps it to (0,1).

$$P(Y = 0|X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

$$P(Y = 1|X) = \frac{1}{1 + e^{\beta_0+\beta_1 X}}$$

The first equation is obtained by taking the inverse of the logit function:

$$log\Big(\frac{p(X)}{1 - p(X)}\Big) = \beta_0 + \beta_1 X$$

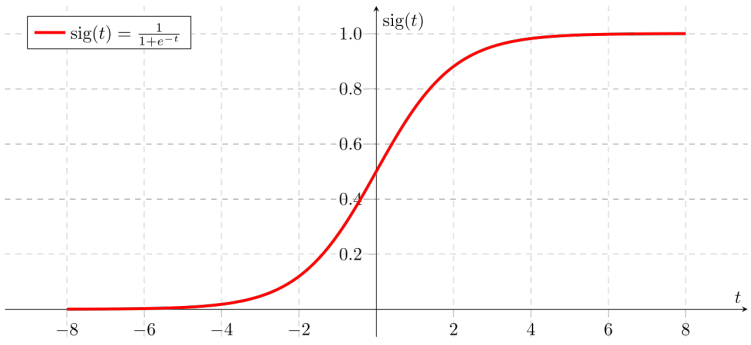


Figure 13: Sigmoid Function

In order to obtain an estimate of the β coefficients, maximum likelihood estimation (MLE) is used.

$$l(\beta_0, \beta_1) = \prod p(x_i) \prod 1 - p(x_i)$$

Once the probabilities are obtained, the classification is made applying a threshold to them, for example is assigned the label 0 if P $\geq$ 0.5.

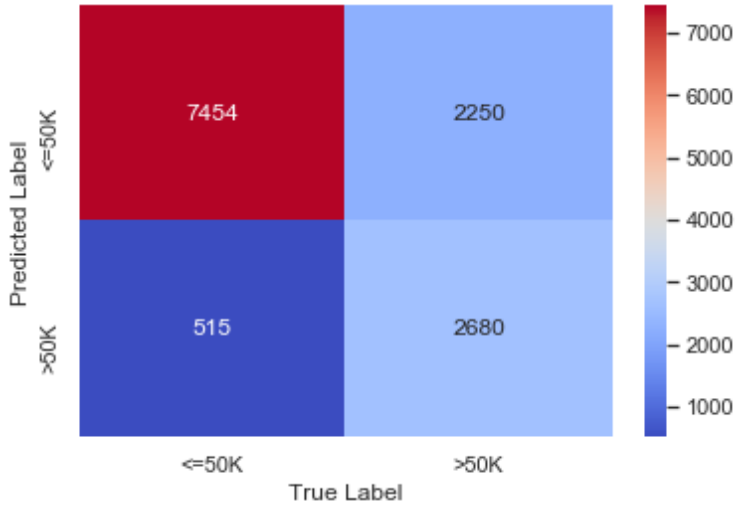


Figure 14: Logistic regression confusion matrix. F1 Score: 0.66

5.4 Support Vector Machine

SVM is a supervised method that solves an optimization problem: Find the hyperplane that maximize the margin between two classes in the feature space. More specifically, the distance is computed between the margin and the closest data point of each class, that are called support vectors.

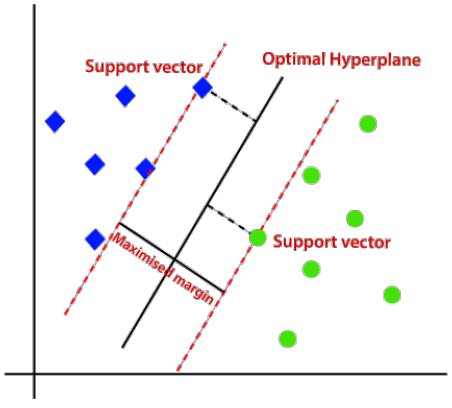


Figure 15: Linear Support Vector Machine

Given an hyperplane

$$f(X) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

We want to find the β that maximize the margin M such that $y_i f(x_i) > 0 \ \forall i$

Often happens that data are not linearly separable solving the hard margin problem. One of the alternative is the soft margin problem: We introduce slacks variable and allow for some mistakes. The hyperparameter C is the cost of misclassification, when C is large, the algorithm tries to maximize the number of points correctly classified while a smaller C will encourage a larger margin at the cost of training accuracy.

$$\min_w \frac{1}{2} w^t w + C \sum_i max(0, 1 - y_i w^t x_i)$$

One of the strongest point of support vector machines is that they can be used with kernels. Kernels are symmetric functions that correspond to a scalar product in an higher dimensional features space. This solution is much more efficient and easier than computing the mapping directly. Hence, the hyperplane is computed in the new space where data could become linearly separable.

$$K(x, x') = < \phi(x), \phi(x') >$$

According to the Mercer Theorem a kernel is valid if it is symmetric and the corresponding gram matrix $G_{ij} = K(x_i, x_j)$ is positive semidefinite. Some of the most popular kernels are rbf (radial basis function) that compute the distance using a gaussian curve, polynomial and sigmoid.

$$K(x, x') = \exp\Big(- \frac{||x - x'||^2}{2\sigma^2} \Big)$$
$$K(x, x') = (1 + < x, x' >)^2$$
$$K(x, x') = tanh(\gamma x^T x' + r)$$

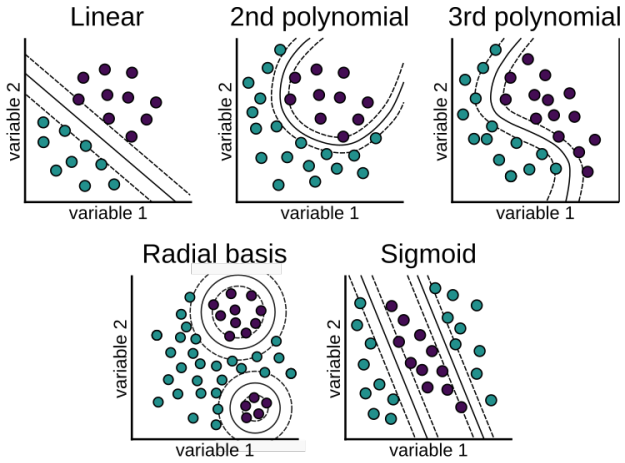


Figure 16: Examples of kernel functions

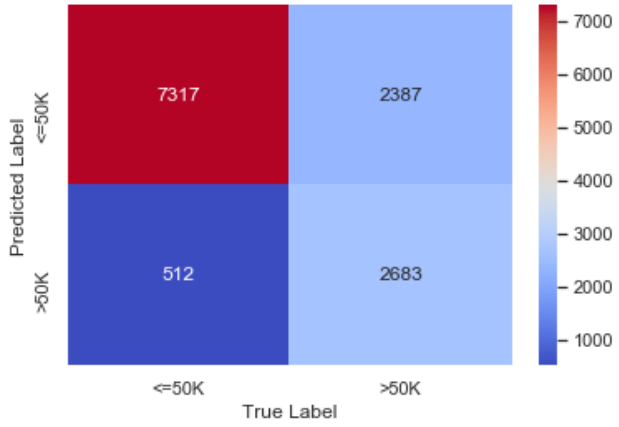


Figure 17: SVM with RBF kernel confusion matrix. F1 Score: 0.64

5.5 K-nearest neighbors

The last algorithm analyzed is K-nn. The reason behind this algorithm is not building an internal general model but storing all the training data with their label and for each query point compute the distances from all the other points. Then the k nearest neighbors are considered and the label is assigned by majority voting. The strong point of K-Nearest Neighbors is that is very easy to implement and to use. On the other hand, it can be very expensive to compute all the distances for large dataset and there may be memory issues. Choosing a small k means that the classification may be more sensitive to outliers, but it's not a good idea making it too big otherwise neighbors that are far apart (maybe irrelevant) will be taken into account.

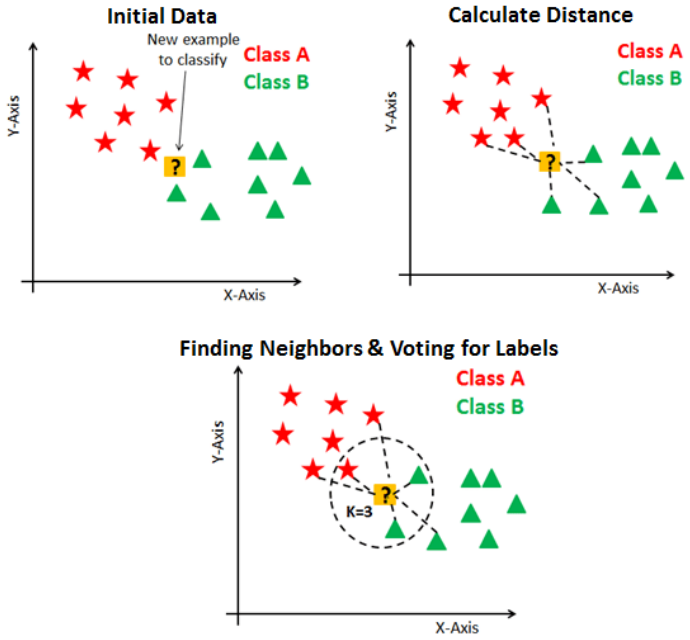


Figure 18: Example of K-nn

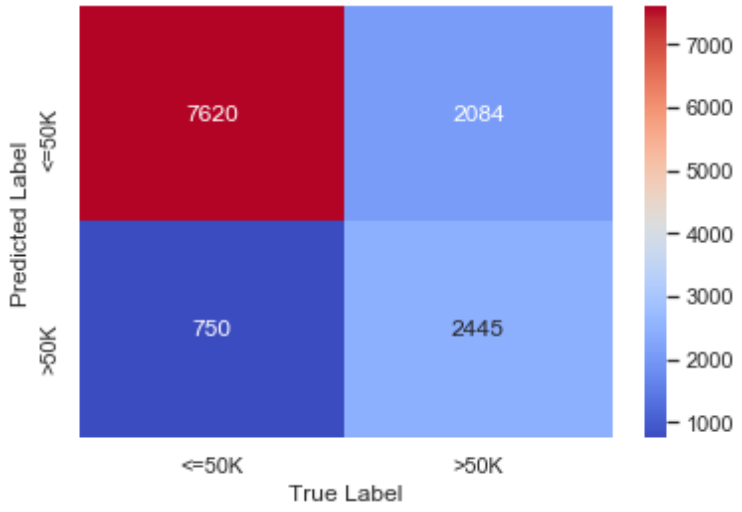


Figure 19: K-nn confusion matrix with k=10. F1 Score: 0.63

6 Conclusions

As we can see from the plots below, with all the models being around 78% in terms of accuracy except for decision tree that is slightly worse. Regarding f1, logistic regression is the best performing model but it is no so far from the others. Notwithstanding the oversampling, there are still a lot of false positive and false negative (as showed in the confusion matrices) and a strong bias towards the majority class $\leq 50K$ is present.

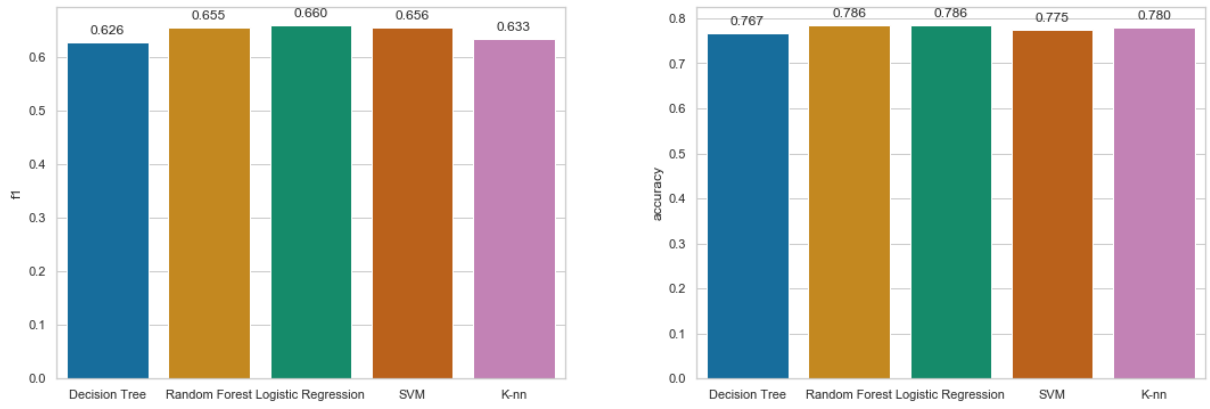


Figure 20: Final Results in terms of f1 (left) and accuracy (right)

For completeness, here are also reported the precision and recall results for each model. As expected, recall scores are much more lower than precision scores, which for some algorithms exceed 80%.

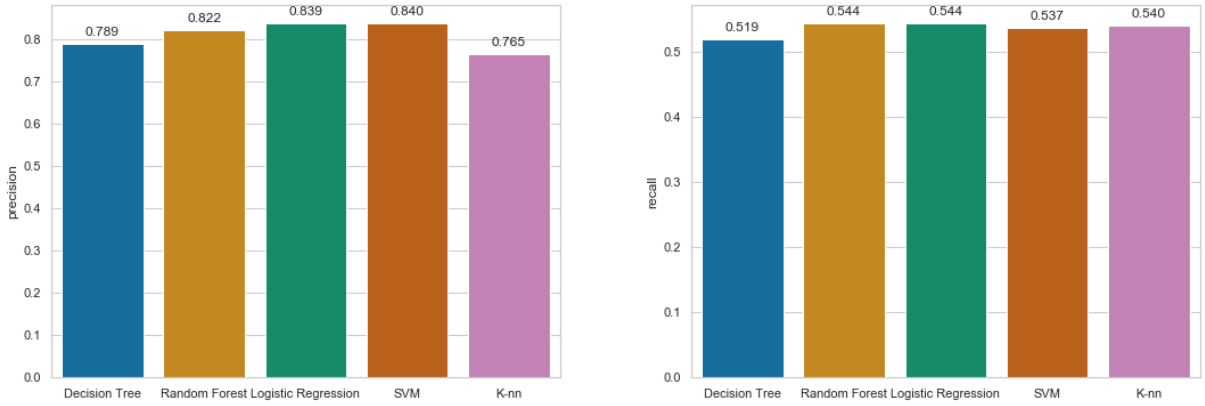


Figure 21: Final Results in terms of precision (left) and recall (right)

7 References

1. Adult dataset: http://archive.ics.uci.edu/ml/datasets/Adult
2. Code available at: https://github.com/mdn33/MathematicsInMachineLearning