Machine Learning Project

Andrea Iglesias, Meritxell Arbiol

June 2022

Abstract

In this project we have applied different models and concepts seen in the course on a dataset to predict the salary given a series of characteristics of several individuals. In this report we have also explained the reasons for each decision we have made and the problems we have had and how we have solved them, either during the preprocessing part or during the different models tested, as well as the final solution we have reached and the conclusions.

1 Introduction

The objective of this project is to predict whether or not a person will make more than $50K given a set of variables. The database that we have used comes from an extract carried out by Barry Becker from the 1994 Census database. From this dataset, we want to know which features are most likely to affect someone's salary and which ones are dispensable from our analysis. Since the problem is to predict when a person earns more than $50K per year, based on census data, we are dealing with a binary classification problem.

The approach that will be used in this project consists of making a series of models using different algorithms. Each of them will have different metrics that will give us a precise view of how our model is performing, a different complexity, and different behavior when generalizing. After analyzing each of the different models, seeing their complexity, their suitability for the data set, and the metrics, a model will be chosen that will be the "best" of the set of developed models.

1.1 OUR DATASET

This database comes from an extract carried out by Barry Becker from the 1994 Census database [1]. Our goal with this dataset is to determine whether a person makes over 50K a year. This dataset consists of 32.561 records and 15 attributes, which are the following ones:

- **age:** numerical variable that indicates the age of the individual.
- **workclass:** categorical attribute that represents the type of employment of an individual. *Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.*
- **fnlwgt:** numerical variable that represents a weight, each weight corresponds to a socio-economic characteristic of the population, therefore, people with similar demographic characteristics should have a similar weight.

[1] Dataset source: *https://www.kaggle.com/datasets/ayessa/salary-prediction-classification*

- **education:** categorical variable that represents the level of education of the person. *Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.*
- **education-num:** numeric variable that represents the previous one.
- **marital-status:** categorical variable that expresses the marital status of the person. *Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.*
- **occupation:** categorical variable that describes the type of profession that the individual has. *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:** categorical variable represents what this individual is relative to other. *Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.*
- **race:** categorical variable that describes the individual's race. *White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.*
- **sex:** categorical variable that corresponds to the sex of the individual. *Female, Male.*
- **capital-gain:** numeric variable that corresponds to the registration of the person's capital gain.
- **capital-loss:** numeric variable that corresponds to the registration of the person's capital loss.
- **hours-per-week:** numerical variable that corresponds to the hours of work per week.
- **native-country:** categorical variable representing country of origin. *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, TrinadadTobago, Peru, Hong, Holand-Netherlands.*
- **Salary (Target):** binomial variable that classifies the salary into two categories. $>50K$, $\leq 50K$.

2 Related Previous Work

There are projects done in the past that have worked with this same dataset[2]. The results obtained in these studies range around 80% recall.

We see that in the data preprocessing part, in our case we give it a great weight and we have made a quite exhaustive analysis compared to the works we have seen published before. In the same way, we have seen that none of the studies deal in any way with outliers.

In addition, concerning missing values, they only check for null values, without taking into account that there might be '?' or '99999' values that should be interpreted as null and treated, otherwise they can affect our predictions.

[2]Link to related previous work: *https://www.kaggle.com/datasets/ayessa/salary-prediction-classification/code*

The predominant models with the highest accuracy were made with the Random Forest and Xboost algorithms. We are going to test several models and compare them with each other to see if we can finally improve the previous results.

3 Data Exploration Process

3.1 Data Exploration

In order to have a first contact with the data, we have carried out a general exploration to familiarize ourselves with the variables.

As we can see in the *Figure 1*, the first variable is *age*, takes values between 18 and 90 years, although most of the individuals in our dataset are between 25 and 45 years old approximately. Regarding the *workclass* variable, we see that there is a clear tendency towards the private sector. We also notice that most of the individuals in our data have *HS-grad* education, followed by *Some-college* and *Bachelors*, and we identify how a very low number of people have *Preschool* level as the highest education level achieved. We also recognize that most of the individuals correspond to white males, from the United States, and with a salary below $50K.

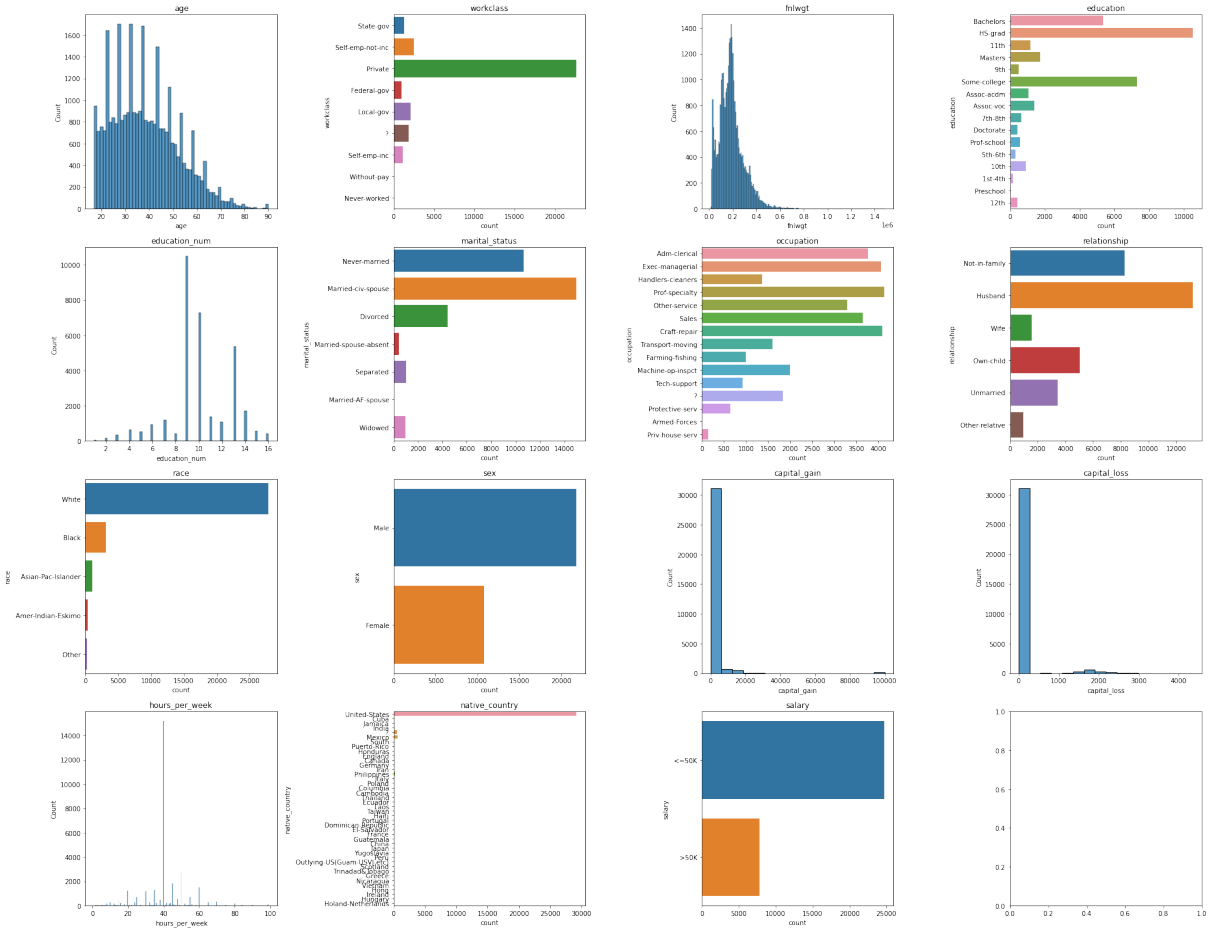

Figure 1: Features exploration of the data

In *Figure 2* we have a general description of our numeric variables, with some statistical values like their maximum, minimum and counts.

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

Figure 2: Description of the numeric features of our dataset

Some values are remarkable, such as *capital_gain* and *capital_loss*, since it seems that most of the values of these variables are 0 except in some cases. The *capital_gain* is the increase in a capital asset's value and is realized when the asset is sold, so when the value is 0 it indicates that the value remains unchanged, and the same applies to *capital_loss*. For the *capital_gain* variable we have only 7.97% of individuals with a value greater than zero, while for the capital_loss variable only 4.76% of records have a value greater than zero. We will see later how we treat these variables and if they are important for our analysis.

We are also quite surprised that there may be people who are working 99 hours per week. This could be a mistake or it could be that there are certain important positions that require continuous work, or we could also be talking about labor exploitation.

Finally, mention that our dataset is quite unbalanced since almost 76% of the individuals have a value of <=50K for the salary variable, and 24% of individuals have a value of >50K in the target variable. This will be taken into account once we want to analyze the performance of our models with the different metrics that we will calculate.

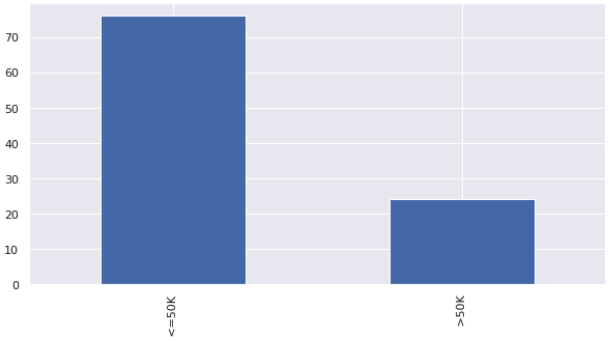

Figure 3: Graph that shows the percentage of samples for each target.

3.2 Pre-processing

3.2.1 Duplicates

In our dataset, we have detected some duplicates, specifically **24** individuals that have no relationship between them. Therefore, we are going to eliminate the duplicate individuals and we are going to keep only 1 copy of each record.

3.2.2 Missing Values

After analyzing if there are missing values in any of our variables, we have seen that there are missing values in 4 of them: *workclass*, *occupation*, *native_country* and *capital_gain*. Even so, the number of missing values for each of the variables is very low.

- The feature *workclass* has 5.64% of missing values.
- The feature *occupation* has 5.66% of missing values.
- The feature *native_country* has 1.79% of missing values.
- The feature *capital_gain* has 0.49% of missing values.

Due to the small number of missing values with respect to the total number of samples we initially had, we have decided to eliminate the lines containing them, going from having a total of 32.537 samples to 29.991. Having reduced the volume of total samples by 7%.

3.2.3 Outliers

For the study of the outliers we are going to analyze only the numerical attributes: *age*, *capital_gain*, *capital_loss* and *hours_per_week*.

We will leave out of the analysis the variables *fnlwgt* (because it is not itself a continuous variable but rather it is like an identifier that is assigned), and *education_num* (it is the same variable as *education* only that the categories have been numbered).

- age

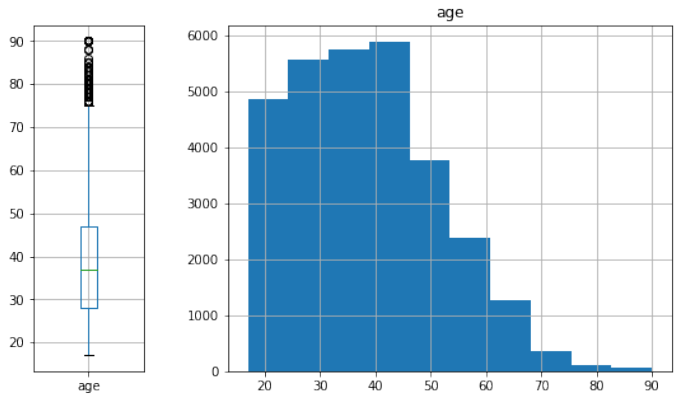

Figure 4: Exploration of the outliers for the variable age.

For the age variable, we have records ranging from 17 to 90 years old. If we set a threshold of 1.5 using the IQR method, we get that 167 individuals between 76-90 years old are outliers.

Let's take a closer look at these individuals. We see that most of them work in the private sector, specifically in the executive or managerial, professional specialty, sales or administration clerical fields, and almost the 85% earn a salary less than 50K.

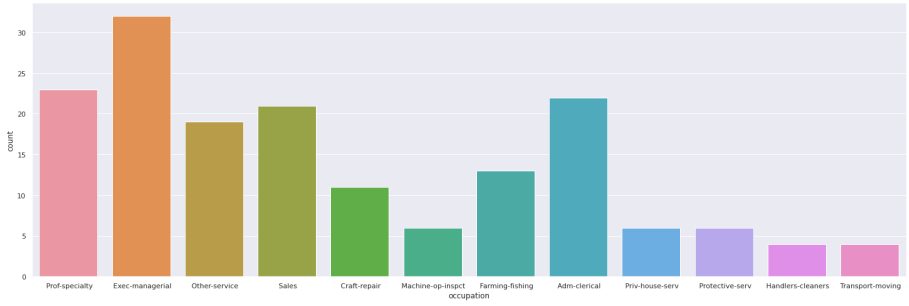

Figure 5: Area of work for the individuals classified as outliers for the variable age.

We are not going to get rid of these outliers since we believe that they are not erroneous data and they can be used to classify these cases in the final model.

- *capital_gain*

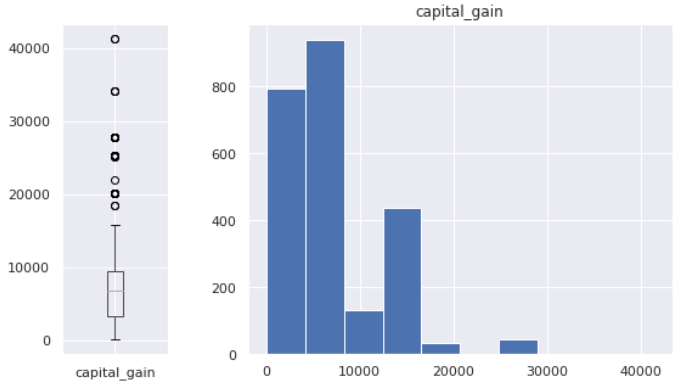

Figure 6: Exploration of the outliers for the variable capital_gain.

Most of the samples in the dataset have the *capital_gain* variable equal to 0. To study the outliers we will ignore all the 0s and do the outlier analysis with the rest. We observe that 87 records are considered outliers with an IQR of 1.5, with a *capitain_gain* between 18481 and 41310.

We are not going to deal with these outliers since, as most individuals have this variable equal to 0, what we are going to do is convert *capital_gain* into a categorical variable that is 1 when *capital_gain* is greater than zero and 0 otherwise.

- *capital_loss*

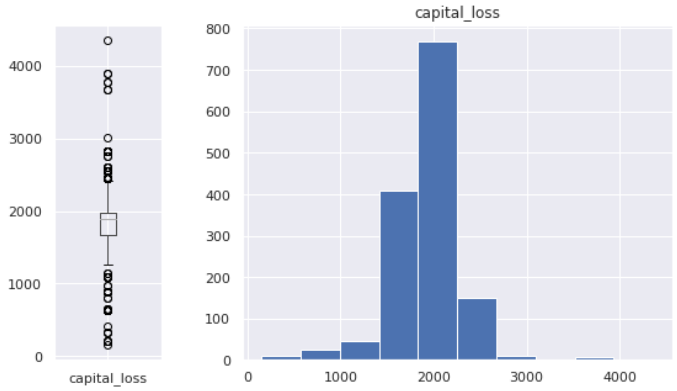

Figure 7: Exploration of the outliers for the variable capital_loss.

Most of the samples in the dataset have the *capital_loss* variable equal to 0. To study the outliers we will ignore all the 0s and do the outlier analysis with the rest.

In this case, we are not going to consider the lower outliers since the variable *capital_loss* can be a value below the mean. Therefore we are only going to look at the upper outliers.

We observe that 53 of the individuals in the sample are considered outliers with an IQR of 1.5, which corresponds to a *capital_loss* between 2444 and 4356. We are not going to deal with these outliers since, as only 4.75% of the records have a *capital_loss* greater than zero, what we are going to do is to remove this variable from the dataset because it does not provide us any additional information for the models we want to create.

- *hours_per_week*

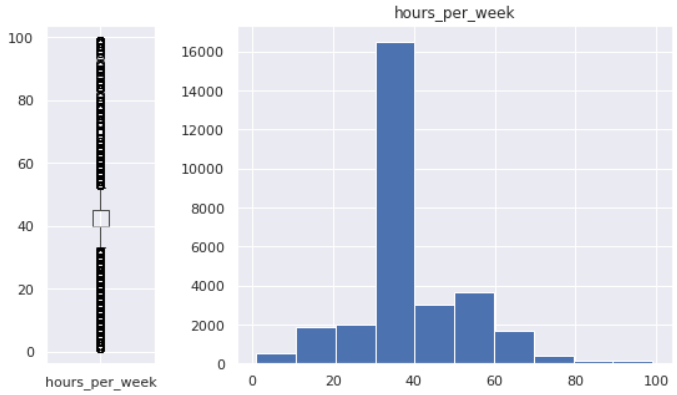

Figure 8: Exploration of the outliers for the variable hours_per_week.

Most individuals in the dataset have the variable *hours_per_week* equal to 40 (full day). Below 40 hours it would be a part-time contract, which is normal, so we will focus our attention on the values of this variable that are well above 40 hours.

We observe that 3270 records are considered outliers with an IQR of 1.5, with the variable *hours_per_week* between 53 and 99.

With a more in-depth analysis of these individuals, we have observed that 58.9% have a salary of less than 50K while 41.1% have a salary of more than 50K. Since these records do not provide us with information about the target, we are going to convert this variable into a categorical variable with 3 categories:

  - part_time (from 0 to 30 hours a week)
  - full_time (from 30 to 50 hours per week)
  - extra_full_time (from 50 hours per week)

3.2.4 Modification and Derivation of New Variables

Another change we have decided to make in our data to facilitate the implementation of the different models we are going to test is to modify the value of the target to 0 or 1 depending on whether we have $<= 50K$ or $>50K$ respectively.

The last thing we have made is to the variable *native_country*, since almost all the values of this variable were United-States, being very few different, we have converted this variable to binary, having only two values: *NOT_United-States* and *United-States*.

4 Modeling

We have tested several different models to find the model that best fits our data and gives a better prediction of whether the salary someone will earn will be above or below $50k. The models we have worked with are the following:

1. Linear Discriminant Analysis
2. Quadratic Discriminant Analysis
3. KNeighborsClassifier
4. Naive Bayes
5. Logistic Regression
6. Random Forest
7. Ensembles

To explain the validation method we have to take into account that the target classes are strongly unbalanced, accuracy is not a good metric choice because it is very sensitive to unbalanced data, therefore we are going to take F1 as a reference metric in our modeling. F1 score is a good way to solve the problem of imbalanced data because it takes into account not only the number of prediction errors that our model makes but also look at the type of errors that are made (*i.e.* F1 is the harmonic mean of precision and recall), while accuracy only measures the number of predictions that have been correctly classified for the total number of predictions. Also, in this particular problem, all the categories are equally important. For this reason, we will use the macro average of our metrics instead of focusing on the metrics of one specific class.

To train the models and finally validate the best of them all, we are going to divide our data into train and test partitions with 10-fold cross-validation over the train partition for deciding hyperparameters. To calculate the metrics and compare the models we have used 10-fold cross-validation metrics.

In addition, we have also applied a little preprocessing to the data before generating the models. This preprocessing consists first of normalizing the data because the different numerical variables show different ranges of values and so that this does not interfere with the results of our models we have decided to scale them. And then we have applied one-hot encoding to the categorical variables since most of our models cannot deal directly with variables of type "Category".

Regarding the Naive Bayes model, we have generated it twice with different data, the first time with the data preprocessed in the same way as mentioned above, and when we saw that the results we obtained were considerably lower (results are shown in the next section) we decided to generate another model with only the numerical data. With this second model, the F1 has increased but still not as much as we expected compared to the results obtained with other models.

We tested the models first with the variable *hours_per_week* by removing the variable called *hours_per_week_interval*, and then the other way around, to see with which of the two variables the models worked better since the two cannot be in the same model due to the high correlation

between them. After the results were obtained, we decided to keep the numerical one because the predictions were higher.

Regarding the *education_num* variable, we have eliminated it since we have the categorical variable called *education*, and we believe that it is the variable that makes the most sense to keep since the numerical variable was the same but the different labels were numbers, and since there were so few different values it did not make much sense to analyze with this variable.

Finally, scikit learn decision trees classifier do not handle categorical attributes then we have to transform them to numerical as we did before for the other models using one-hot encoding. However, it is a tree-based model and hence does not require feature scaling.

5 Results

In this section, we have the results obtained with each of the models used.

5.1 Linear Discriminant Analysis (LDA)

LDA has a closed-form solution and therefore has no hyperparameters.

	Accuracy	**F1 Macro**	**Precision Macro**	**Recall Macro**
LDA	0.835	0.763	0.785	0.747

5.2 Quadratic Discriminant Analysis (QDA)

For this model, we have tested several regularization parameters to know which one is the most optimal for our case, and finally, the one that gives us the highest F1 is with the parameter `reg_param=0.1`.

	Accuracy	**F1 Macro**	**Precision Macro**	**Recall Macro**
QDA	0.773	0.738	0.728	0.794

5.3 KNeighborsClassifier

We used the function GridSearchCV to tune the following hyperparameters:

- n_neighbors: *range(20,50,2)*,
- metric: *euclidean, minkowski, manhattan*,
- weights: *uniform, distance*

Using the metric F1 as a reference we got that several sets of parameters have a similar result. Among them we have chosen `n_neighbors=48`, `metric='minkowski'`, `weights='distance'`, and these are the results obtained:

	Accuracy	**F1 Macro**	**Precision Macro**	**Recall Macro**
KNN-48	0.828	0.755	0.771	0.744

5.4 Naive Bayes

	Accuracy	**F1 Macro**	**Precision Macro**	**Recall Macro**
NB	0.567	0.562	0.659	0.694
NB-num	0.769	0.574	0.688	0.573

5.5 Logistic Regression

We used the function GridSearchCV to tune the following hyperparameters:

- C: *np.logspace(-3,3,7)*,
- solver: *newton-cg, lbfgs, liblinear, sag, saga*,
- penalty: *none, elasticnet, l1, l2*

Using the metric F1 as reference we got that several sets of parameters have a similar result. Among them we have chosen `C=10`, `penalty='l2'`, `solver='saga'`, `multi_class='multinomial'`, because for large datasets, *'sag'* and *'saga'* are faster. These are the results obtained:

	Accuracy	**F1 Macro**	**Precision Macro**	**Recall Macro**
Logistic Regression	0.837	0.764	0.787	0.749

5.6 Random Forest

We first try a standard decision tree. We have optimized it with this hyperparameters:

- criterion: *gini, entropy*,
- max_dephts: *None, 5, 10, 15, 20*,
- min_samples_split: *2, 3, 4, 5*,
- min_samples_leaf: *1, 2, 3, 4, 5*
- max_features: *auto, sqrt, log2, None*

The time it took to train this model using 10-folds cross-validation was: `0:06:05.550599`. Finally, the best hyperparameters are `criterion='entropy'`, `max_depth=None`, `max_features='auto'`, `min_samples_leaf=5`, `min_samples_split=3`.

Then we tried the RandomForestClassifier with the default parameters and then we tunned all these hyperparameters:

- n_estimators: *200, 300*,
- max_depth: *10, 50, 100*,
- min_samples_split: *2, 5, 10*
- min_samples_leaf: *2, 4, 5*,
- class_weight: *None, balanced_subsample*,
- criterion: *gini, entropy*
- bootstrap: *True, False*

The time it took for this model to use 10-folds cross-validation was: `1:32:21.768429`. Finally, the best hyperparameters are `bootstrap=True`, `class_weight='balanced_subsample'`, `criterion='entropy'`, `max_depth=100`, `min_samples_leaf=2`, `min_samples_split=2`, `n_estimators=300`.

Moreover, we applied the ExtraTreesClassifier with the default parameters, and then we optimized it using the same hyperparameters that we used in Random Forest. The time it took this model using 10-folds cross-validation was: `1:01:29.212688`. Finally, the best hyperparameters are:

	Accuracy	F1 Macro	Precision Macro	Recall Macro
DT	0.816	0.737	0.754	0.725
RF_default	0.828	0.754	0.771	0.742
RF_CV	0.821	0.777	0.762	0.803
extra_trees_default	0.815	0.741	0.751	0.734
extra_trees_CV	0.804	0.766	0751	0.808

We see that the one that gives us the best results is the model that corresponds to the Random Forest using 10-fold cross-validation. If we analyze this model better, we see that the attributes that contribute the most in terms of prediction are the following: *age*, *martial_status* (married), *fnlwgt*, *hours_per_week*, *martial_status* (never married). The top 10 can be seen in the following figure. On the other hand, the 5 variables that contribute the least to the model are: *marital_status* (married AF spouse), *occupation* (private house service), *education* (preschool), *workclass* (without-pay), *occupation* (Armed-Forces).

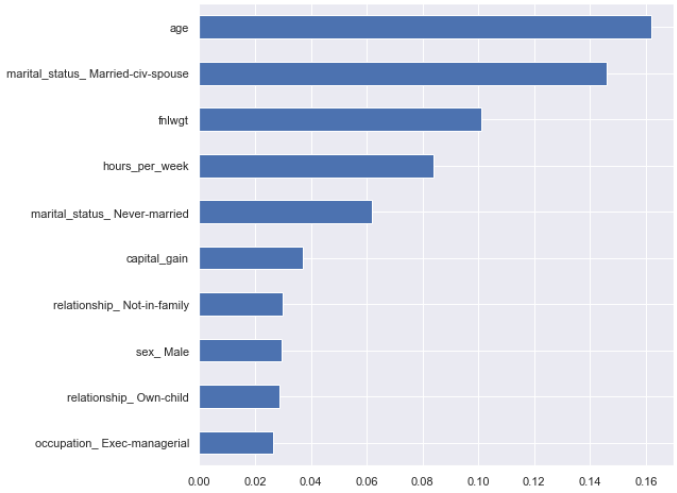

Figure 9: Top 10 most important variables for predicting the target.

5.7 Ensembles

Here is a summary of the results obtained so far:

	Accuracy	F1 Macro	Precision Macro	Recall Macro
LDA	0.835	0.763	0.785	0.747
QDA	0.773	0.738	0.728	0.794
KNN-48	0.828	0.755	0.771	0.744
NB	0.567	0.562	0.659	0.694
NB-num	0.769	0.574	0.688	0.573
Logistic Regression	0.837	0.764	0.787	0.749
DT	0.816	0.737	0.754	0.725
RF_default	0.828	0.754	0.771	0.742
RF_CV	0.821	0.777	0.762	0.803
extra_trees_default	0.815	0.741	0.751	0.734
extra_trees_CV	0.804	0.766	0751	0.808

Now that we have a set of models we are going to combine them into more powerful classifiers. We will combine the 4 top best performance models regarding the F1 score, that correspond to LDA, Logistic Regression, RF_CV and extra_trees_CV.

With a voting classifier, we can make the models vote in a "hard" fashion (majority vote) or "soft" fashion (averaging probabilities). We are making a voting classifier with our best models. We can see that soft voting works better. We can also see that the models are performing better together.

Another ensamble we can use is a Stacking classifier. It will train another classifier on top of the result of our classifiers instead of voting.

Below it can be found all the combinations we have tested with their respective metrics:

	Accuracy	**F1 Macro**	**Precision Macro**	**Recall Macro**
stacky_RF_ET	0.836	0.766	0.784	0.753
stacky_RF_LDA	0.839	0.771	0.787	0.759
stacky_RF_logreg	0.836	0.767	0.784	0.754
voting_soft_RF_ET	0.814	0.773	0.757	0.807
voting_hard_RF_ET	0.825	0.779	0.765	0.802
voting_soft_RF_LDA	0.839	0.781	0.784	0.778
voting_hard_RF_LDA	0.838	0.762	0.793	0.743
voting_soft_RF_logreg	0.827	0.739	0.780	0.718
voting_hard_RF_logreg	0.838	0.7618	0.793	0.743

Finally we can observe how the models improve with respect to the F1 metric and the one with this highest measurement corresponds to the combination of the Random Forest with the best hyperpameters together with Linear Discriminant Analysis through soft voting.

6 Final Model

After testing many models and spending a lot of time discovering which hyperparameters give the best results, it is time to choose our final model.

The best model we have trained according to our metrics is the voting soft classifier with the best random forest and linear discriminant analysis. Now let's check our metrics over this model.

Our test results are very similar to the validation ones. This means that our model would probably generalize well if we used it on new data.

	Precision	**Recall**	**F1-score**	**Support**
0	0.836	0.89	0.90	7466
1	0.68	0.67	0.67	2432
accuracy			0.84	9898
macro avg	0.79	0.78	0.78	9898
weighted avg	0.84	0.84	0.84	9898

We can also see how, as it is an unbalanced database, the model better predicts individuals with target $<=50K$ than those with target $>50K$. Below we can see the confusion matrix in which it shows us that 11.51% of individuals with target $<=50K$ have been erroneously predicted and instead 49.20% of records have been erroneously predicted with target $>50K$.

	Predicted 0	**Predicted 1**
Target 0	6695	771
Target 1	802	1630

7 Conclusions

After testing several models on the previously preprocessed data and comparing the results obtained between them, we have seen that all models (except Naive Bayes) have an F1 of around 75%, so the results are quite similar.

We have also decided to combine the best performing models to see if we were still able to improve the result a bit more, and we have done so: LDA and Random Forest make a good combination, and allow us to improve the classification a bit more reaching an F1-score of 78%. We have also seen that it is a model that generalizes very well since the metrics with the train data and with the test data are very similar.

A possible extension of this project could be to try to apply a multi-layer perceptron model, given that logistic regression has given us good results, by correctly training this algorithm we may get better results. In our case, we have not considered this model since the data is quite simple and we believe that applying a multi-layer perceptron is to include too much complexity.

This work has helped us to understand how important the initial data cleaning is, as well as to understand what each algorithm does and with which parameters it will work best.