Data Challenge Presentation Summary

Jean Barré

April 3, 2022

1 Method and tools

We based our work on the python programming language and the libraries build on top of it. For the data analysis and manipulation we used Pandas and Numpy, for visualisation we used Matplotlib and Seaborn. We will discuss further feature engineering and data transforming (see section 3), for that purpose we used the Datetime and the Dirty_cat libraries. We used Scikitlearn wich gives efficient tools for predictive data analysis. Scikitlearn is quite simple to use and it implements state of the art machine learning algorithms.

2 Data Visualization

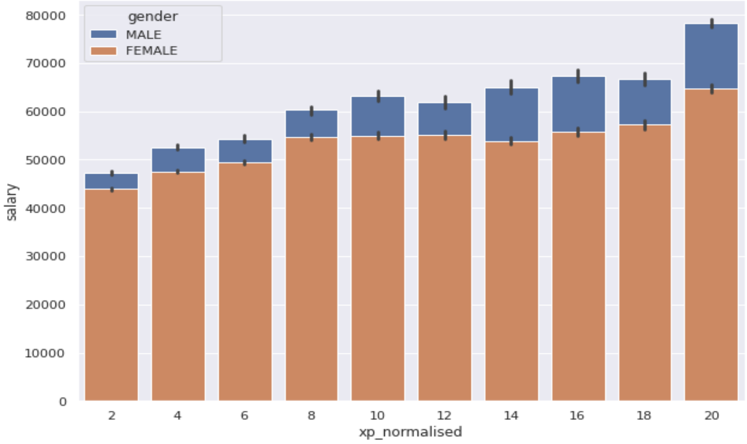

Figure 1: wages according to experience in years with gender differentiation

The dataset was very homogeneous, which is why we consider that the most insightful visualisation was using simple bar-plots rather than more complex ones. The figure 1 shows the wages according to experience in two years period with gender differentiation. We explain how we computed experience in section 3.

We found that there was no strong correlation between experience and the level of wages, since many people earn good wages without any long experience, but this chart seems to support the fact that experience and gender matters in the salary expectation. We can see that the more experience you have the more likely your salary goes up. If you are a male the relationship is even stronger. But the relation is not so obvious and other important factors matter as we will see further.

3 Feature engineering

Before addressing the modelisation step, we had to pre-process our data to change string labels from our categorical columns into a representation that is more suitable for a model. An encoder is needed to turn a categorical column into a numerical representation. Since the order of values in our columns did not matter, we used the one-hot encoding procedure rather than the Ordinal encoding one to transform the data from string labels to numerical values.

On top of that, we had two main issues with the columns of the dataset of this challenge. First of all, there was a column called "hire_date" wich was nice but it did not have an understandable format for our algorithms. We decided to retrieve the 'experience' of a worker as a number of days since they were hired. We chose the number of days because it was more precise than the years scale for example. We handled the issue using the Datetime python library. We used the Standard-Scaler procedure to have a normally distributed column.

Then, a challenge was to handle the column 'class_title'. The one-hot encoder is actually not well suited, as this columns contains 1383 different entries. Indeed it has two main drawbacks: for high-cardinality variables the dimensionality of the transformed vector becomes unmanageable. Besides, the mapping is completely uninformed: "similar" categories are not placed closer to each other in embedding space. We thought that the 'class_title' column had a significant impact on the salary so we look for a way to transform this column correctly.

We implemented the python library named dirty_cat to handle this issue. This library gives encoders robust to morphological variants. We decided to use the similarity encoder which is based on calculating the morphological similarities between the categories and the gap encoder which can be understood as a continuous encoding on a set of latent categories estimated from the data. The main advantage of gap encoding is that it makes the dirty features quite understandable for the machine learning algorithm but also for the human analysis of feature importance.

4 Model description and evaluation

The data challenge was about predicting incomes of people based on real life features such as gender or ethnicity. Predicting a continuous value is a well-known problem in machine learning and is addressed with regression tools. We had a supervised learning setting that is to say we trained a model on a train set with a target column i.e. the salary. Once trained, the model has to predict the salary based on all the other features. We evaluated the model performance on the test set.

We used a Scikitlearn pipeline with different encoders (see section 3) and one estimator. For this task we chose the ensemble regressor class with the Random Forest and the Hist-Gradient Boosting algorithms. They are supervised learning algorithm based on the ensemble learning method and Decision Trees. They are good estimators facing heterogeneous data, and that was our case with our dataset.

We evaluated our models thanks to different metrics. The R2 score provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of variance. The mean absolute error and the mean squared error are two ways of evaluating the error made while predicting a continuous value such as a salary. Thanks to them we measure how far the predictions were from the true values. We compare the results obtained in those three metrics to evaluate our models. Moreover, we implemented a cross-validation setting to avoid over-fitting and to ensure the reliability of our models.

Here are the results obtained with the R2, Mean absolute error (MAE) and the root mean squared error (RMSE) metrics in cross-validation for the tree ensembles we computed, the Random Forest (RF) estimator and the Hist-Gradient Boosting (HGB) one, with different encoding strategies for the class_title column, the one-hot encoding (OHE), Gap Encoder (GE) and Similarity encoder (SE). We got the best predictions with the Random Forest estimator with the similarity encoder for the 'class_title' column.

Model Pipeline	R2 score	MAE	RMSE
HGB & GE	88.8%	$5203	$9207
HGB & SE	94.6%	$3534	$6363
RF & GE	91.2%	$3614	$8128
RF & **SE**	**95.0%**	**$2944**	**$6189**

Table 1: Best results