Machine Learning - Project 15

Grace Beyoko and Arina Agaronyan

January 2025

Contents

1 Introduction

The aim of this project is to predict the target variable in the most accurate manner. This report details the methodology followed, from preprocessing the data to selecting the best model and evaluating its performance.

2 Data Cleaning

2.1 Data Overview and Simplification

The merged dataset, called **learn_data**, used in this study was constructed by selectively merging many of the provided datasets. Prior to merging the different datasets, we did some simplification. In fact, after looking at our data, we noticed that many variables were categorical, with a high number of distinct values which could later put strain on the computational power required.

First, the **learn_sport** dataset was created by merging the **learn_dataset_sport** with **code_Sports** by the "Code", and replacing said "Code" by the "Sports_Category", which only takes 5 distinct values instead of the 96 in the original dataset; later, a 6th category was created (a '0' value) to account for all missing value observations.

Similarly, the department codes in **learn_dataset_job** and **learn_dataset_retired_jobs** were replaced with the corresponding regions by merging with **departments**, thus reducing the number of distinct values from 429 to 13. This was also later done again with the "INSEE_CODE" (after merging with **city_adm** to join the corresponding department and then region) to create a regional variable for the current living arrangements - "CURRENT_REG". Additional datasets have been merged to incorporate potentially relevant variables, including **city_loc** and **city_pop** for WSG 84 system GPS coordinates of the cities of residence and their populations respectively.

To simplify work descriptions, individuals were mapped to "N1" categories using the **code_work_description_map**. However, upon closer inspection, the "N1" categories were identical to those of the "JOB_42", and thus the "work_description" variable was removed entirely. Furthermore, the economic sector variable was simplified into a 10 category generalised nomenclature using an externally sourced mapping[1]. These transformations significantly reduced the dimensionality of the data while retaining its informational quality.

The "HIGHEST_CREDENTIAL" and "act" variables were factorised for numeric representation. Similarly, the "sex" boolean variable was transformed into a binary integer column, and most other non-categorical variables were transformed into integer type columns for ease in later predictive analysis. In addition, the "HOUSEHOLD_TYPE" was manually factorised so as to not differentiate between female and male single and mono-parental units, as this distinction already exists within the "sex" variable.

After merging all the complementary datasets with the main ones, a function was applied to merge seemingly duplicate columns (like those with _x and _y suffixes) wherein similar data from employed and retired persons was merged to exist within one column.

[1]Version anglaise des intitulés de la nomenclature agrégée - NA, 2008, at `https://www.insee.fr/fr/information/2028155`

Finally, The string prefixes 'tr_', 'ct_', and 'csp_' for the "employee_count", "Employer_category", and "work_description" variables respectively were also stripped for an integer data type.

2.2 Handling Missing Values

Missing values presented a significant challenge in the dataset. To address this, we began by combining similar columns, the only difference being the group of interest (employed people or retirees). For example, the "Pay" column refers to the income of working individuals, and as a result, retirees had missing values in that column, even though they received income in the form of a retirement pension ("Retirement_pay"). We resolved this by merging the two categories and assigning a value of '0' for individuals who had never worked or were unemployed. Similarly, missing values in the "WORKING_HOURS" column were set to '0' for unemployed individuals.

Next, to handle the non-imputable missing data in job and retirement related categorical variables, two new categories - 'Unemployed' and 'Employed' - were created and selectively applied to these instances. For example, 'Unemployed' was set as a value for all missing values in the "TYPE_OF_CONTRACT" column, meaning that either that person had a "JOB_42" value in the 8th category (implying unemployment) or an "act" of 'TACT1_2' (Chômeurs - unemployed); 'Employed' was then set for the remaining missing values in the 3 retirement related variables such as "FORMER_JOB_42", under the condition that "act" is equal to 'TACT1_1' (Actifs ayant un emploi, y compris sous apprentissage ou en stage rémunéré - actively employed).

Prior to imputation and other measures, there were 9 variables with missing values (see Table 2).

In order to understand the missingness of the variables, we analyzed the proportion of missing variables by JOB_42 groups, the percentage of missing value per category and the missingness correlation of those variables.

Variable	**Missing Values**	**% of Total Values**
Employer_category	8317	16.60
employee_count	8129	16.20
WORKING_HOURS	6957	13.90
Pay	6939	13.90
TYPE_OF_CONTRACT	6939	13.90
WORK_CONDITION	6939	13.90
labor_force_status	6939	13.90
Economic_sector_num	6939	13.90
FORMER_REG	2418	4.80
JOB_REG	2418	4.80

Table 1: Learn Data Missing Values.

For some of these, when grouped by "JOB_42" category, the percentage of total missing values was 100%, and thus meaningful imputation was not possible. As such, we once again relied on external sources such as **INSEE** and **URSSAF** and manually applied an average value for the "Pay" and "WORKING_HOURS" for these select "JOB_42" types.

Additionally, we investigated the rows with only 1 or 2 missing observations for meaningful imputation - as partly illustrated in Table 2. Regarding the rows with only 1 single missing value, implying a completely at random missingness, there were 4 main variables - "JOB_REG", "FORMER_REG", "Employer_category" and "WORKING_HOURS". For the regional columns, we replaced the NA in one with the value of the other, thus assuming that the former region and the region of the former job (as this was specifically for the retirees) was the same, using evidence of the majority (consistently over 70%) of the other non-missing observations within the respective "JOB_42" categories showing that trend. Additionally, following that same assumption, we replaced all other missing values in "JOB_REG" with those in "CURRENT_REG".

JOB_42	**Count**	**Total**	**Percentage**	**Missing Value Column**
csp_6_8	38	801	4.70	Employer_category
csp_5_3	13	570	2.30	WORKING_HOURS, Employer_category
csp_7_8	63	3375	1.90	Employer_category
csp_6_7	19	1319	1.40	Employer_category
csp_6_5	4	422	0.90	Employer_category
csp_6_4	5	706	0.70	Employer_category
csp_7_7	21	4288	0.50	Employer_category, WORKING_HOURS
csp_5_6	7	1935	0.40	Employer_category
csp_3_3	2	481	0.40	WORKING_HOURS
csp_6_2	4	929	0.40	Employer_category
csp_5_5	3	1256	0.20	Employer_category
csp_6_3	3	1376	0.20	Employer_category
csp_5_4	4	1642	0.20	Employer_category
csp_4_7	2	1115	0.20	Employer_category
csp_7_5	5	2757	0.20	Employer_category, WORKING_HOURS
csp_4_2	1	862	0.10	Employer_category
csp_4_6	3	2357	0.10	Employer_category
csp_7_4	1	1220	0.10	Employer_category
csp_4_3	2	1441	0.10	Employer_category
csp_5_2	1	2354	0.00	Employer_category

Table 2: Learn Data columns with 1 Missing Value per row by JOB_42.

For the "Employer_category", we used a KNN algorithm for imputation, and when the missing values were in "WORKING_HOURS", we replaced by the median of the given "JOB_42" group. Here, we choose the median due to the negatively skewed nature of the distribution of the variable.

When we had two missing values, we again used a KNN algorithm. At the end, we were left with rows with 9 or 11 missing values. After a concise analysis of the missingness correlation, we realized that those variables are almost always missing together. Due to this high correlation, we chose to apply a MICE algorithm to impute those values.

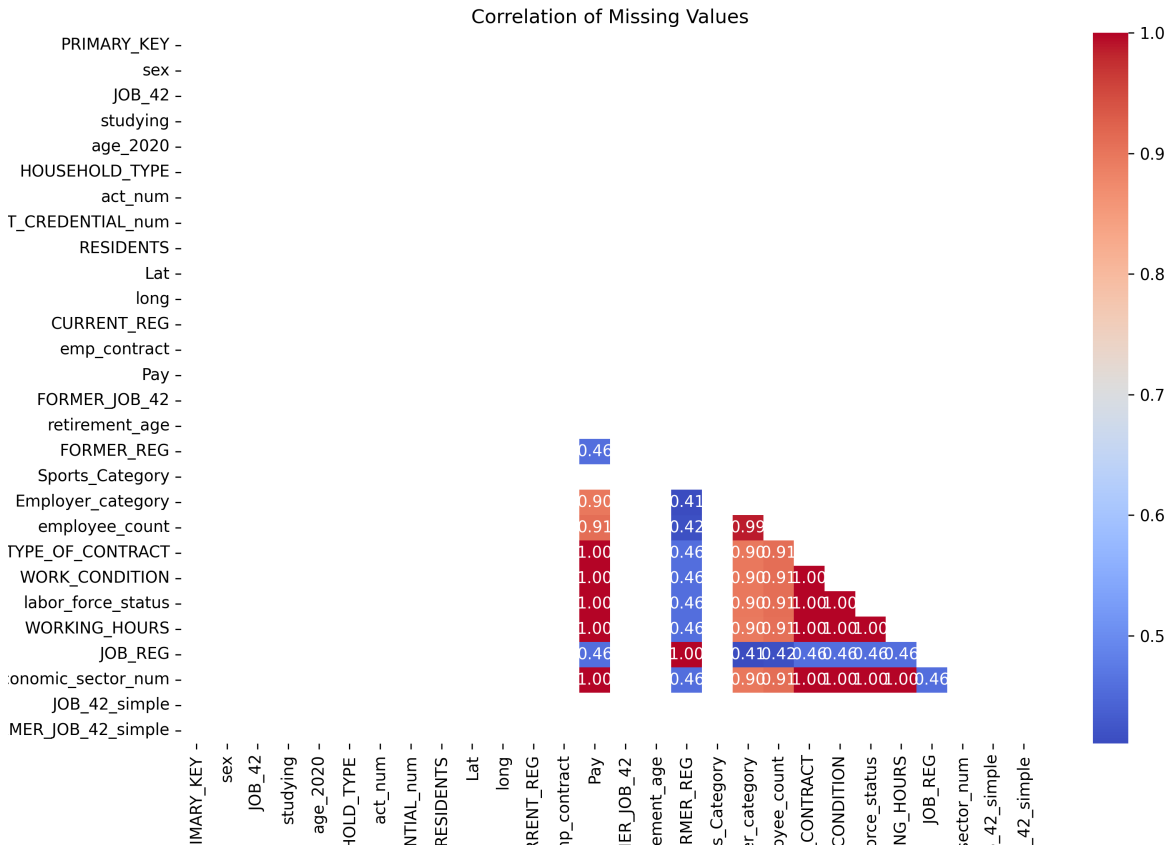

Figure 1: Correlation map of missing variables

Finally, unnecessary variables that did not contribute meaningfully to the prediction or were repetitive were dropped.

3 Prediction

After completing the data cleaning, we chose to apply two different predictive approaches. First, we used a straightforward method, linear regression, followed by a more advanced model, the random forest regressor.

3.1 Linear regression

In order to be able to use a linear regression model, all variables had to first be transformed into numerical types. Except for a few categories including "HIGHEST_CREDENTIAL" and "employee_count" which are ordinal, all other categorical variables were processed by one-hot-encoding. In order to optimise the model for our data, we performed a grid search and 5 folds with a fit_intercept=False selected meta-parameter.

3.1.1 Model Evaluation

The Linear Regression model was evaluated by splitting the encoded **learn_data** into a training set and a test set, with its performance summarised in Table 3. The model achieved an R^2 score of 0.61 on the training set and 0.62 on the test set,

indicating consistent performance and minimal overfitting. The Root Mean Squared Error (RMSE) was 1.28 for both sets, further confirming the model's stability across the data splits.

	Train	**Test**
R^2	0.61	0.62
RMSE	1.28	1.28

Table 3: Learn data train-test error metrics for Linear Regression.

Additionally, the scatter-plot in Figure 1 demonstrates a fairly strong correlation between the predicted and true target values, with points aligning closely around the diagonal. However, by looking at this plot it appear that the model struggles to predict extreme values. Despite its simplicity, the Linear Regression model provides a solid baseline for this task. However, the moderate R^2 score suggests that a more complex model might better capture the underlying patterns in the data.

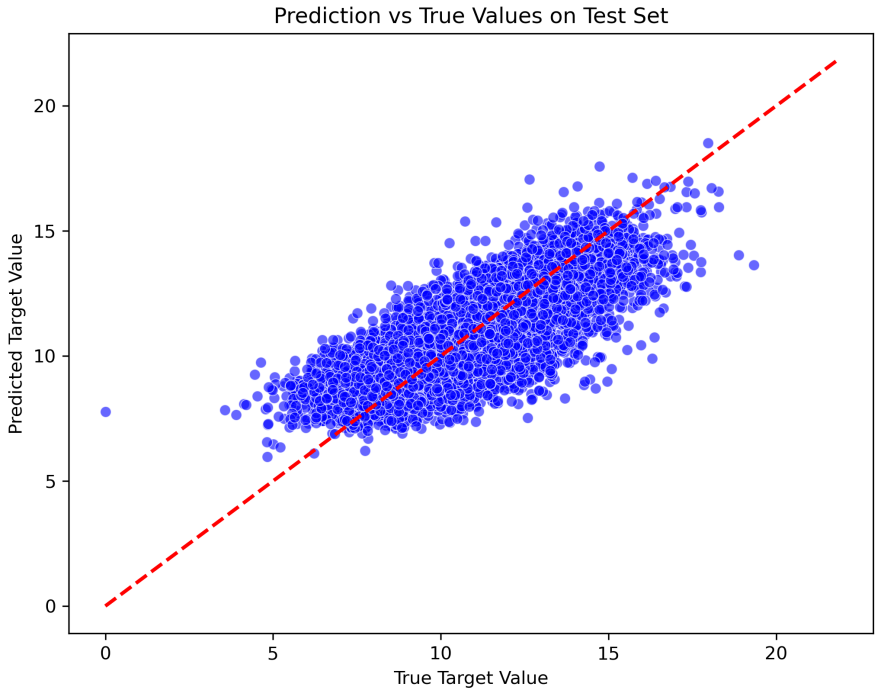

Figure 2: Linear Regression Predictions vs True target values of the learn data

3.2 Random forest

3.2.1 Hyperparameter Tuning

A grid search method was employed to optimise the meta-parameters of the random forest model. This involved setting a random state for reproducability and evaluating the model's performance across various combinations of these meta-parameters:

- Maximum depth: 5, 10, 30, 50
- Minimum samples split: 5, 10, 20

To ensure robust performance evaluation, K-fold cross-validation with 5 folds was used during the grid search, allowing the model to be tested on different subsets of the data. The random forest model with the best-performing meta-parameters was selected as the final model. The optimal configuration included max_depth=50 and min_samples_split=5.

3.2.2 Model Evaluation

The Random Forest Regressor demonstrated strong predictive performance, as we can see in Table 4. In fact, the model achieved an R^2 score of 0.95 on the training set, indicating excellent fit to the training data, and 0.77 on the test set, while clearly implying overfitting, showcases its ability to generalise effectively. The Root Mean Squared Error (RMSE) was 0.44 for the training set and 0.99 for the test set, reflecting lower prediction errors compared to the Linear Regression model.

	Train	**Test**
$\mathbf{R^2}$	0.95	0.77
RMSE	0.44	0.99

Table 4: Learn data train-test error metrics for Random Forest.

As shown in Figure 2, the scatter-plot of predicted versus true target values illustrates that the predictions are closer to the diagonal line, indicating higher accuracy and reduced spread compared to the simpler Linear Regression model. However, similar to the linear regression model our model struggles to predict the more extreme values. Overall, the results confirm that the Random Forest model more effectively predicted the target variable, making it a better choice for our task.

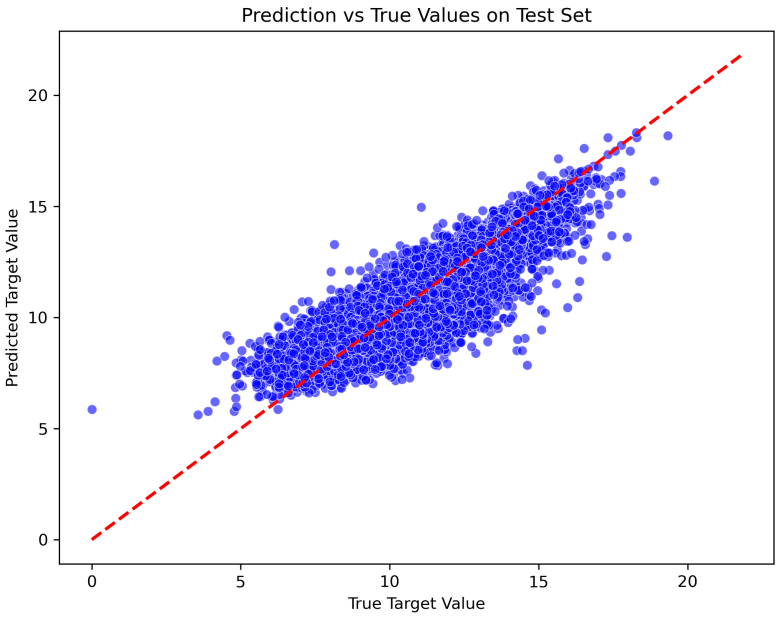

Figure 3: Random Forest Predictions vs True target values of the learn data

In addition to these metrics, an analysis of variable correlation and importance to the target was conducted to identify the most influential predictors. "studying" was clearly the most important variable for the target. Here are the full results:

Variable	target
studying	0.47
WORK_CONDITION	0.29
labor_force_status	0.27
Economic_sector_num	0.25
FORMER_REG	0.22
retirement_age	0.22
TYPE_OF_CONTRACT	0.21
FORMER_JOB_42	0.20
employee_count	0.18
emp_contract	0.16
act_num	0.12
JOB_42	0.10
Employer_category	0.07
Lat	0.05
Sports_Category	0.03
JOB_REG	0.02
HOUSEHOLD_TYPE	-0.02
CURRENT_REG	-0.02
PRIMARY_KEY	-0.03
long	-0.10
HIGHEST_CREDENTIAL_num	-0.18
sex	-0.18
Pay	-0.28
WORK_HOURS	-0.28
age_2020	-0.32

Table 5: Correlation

Feature	Importance
studying	0.23
HIGHEST_CREDENTIAL_num	0.09
HOUSEHOLD_TYPE	0.09
JOB_42	0.08
emp_contract	0.08
sex	0.07
act_num	0.06
PRIMARY_KEY	0.05
long	0.05
age_2020	0.05
Lat	0.04
Pay	0.02
WORKING_HOURS	0.02
retirement_age	0.01
FORMER_JOB_42	0.01
Sports_Category	0.01
employee_count	0.01
Economic_sector_num	0.01
FORMER_REG	0.01
JOB_REG	0.01
CURRENT_REG	0.00
Employer_category	0.00
WORK_CONDITION	0.00
TYPE_OF_CONTRACT	0.00
labor_force_status	0.00

Table 6: Feature Importance

4 Conclusion

To conclude, in this project we tackled the challenge of cleaning a dataset with a significant amount of missing data and then applied machine learning models to predict the target variable. To predict the target variable, we trained two models: a Linear Regression as a straightforward baseline and a Random Forest Regressor for a more robust, non-linear approach. Among these, the Random Forest Regressor outperformed the Linear Regression model, achieving an R^2 score of 0.77 on the test set. Based on this result we expect the R^2 of our final predictions to be around 0.7.