# Predictive Analysis

Sacha Cohen & Loup Theuret

February 2021

## 1 Building the model

The purpose of this report is to build a predictive model for the Emolument given the other variables. The augmented learn.csv will help us to construct the model that will be applied on test.csv. After cleaning the data, we completed the data sets and selected the columns to include in the analysis. As the preparation needed on the data sets before applying machine learning algorithms consisted only in dropping the "Emolument" column and in preprocessing, we chose not to use pipelines.

## 2 Model used

As we want to predict a value and not a class, regression models seems a natural way to proceed.

### 2.1 Linear Regression

The linear regression [1] defines a linear relationship between the dependent variable (emolument) and the independent variables which are the features of the data set. This relationship is of the form :

$$y = b_0 + b_1 x_1 + b_2 x_2 ... + b_n x_n + \epsilon \tag{1}$$

Where $b_1, b_2 ... b_n$ are the regressor coefficients and $\epsilon$ is the error term. The linear regression computes the estimators of the regression coefficients. We want to estimate the closest predictive value f(x) as possible to the true value y. The estimated regression function is defined as :

$$f(x) = b_0 + b_1 x_1 + b_2 x_2 ... + b_n x_n \tag{2}$$

The residual is the difference between the true linear relationship and the estimated regression. The algorithm minimizes this difference by minimizing the sum of squared residuals for each observation (SSR) $\sum_{n=1}^{i} (y_i - f(x_i))^2$. Linear regression is the simplest regression to use and thus is a good starting point.

## 2.2 Ridge regression

Ridge regression is close to linear least squares with l2 regularization. In other words, it adds "squared magnitude" of coefficient as penalty term to the loss function. The loss function is then defined as :

$$\sum_{n=1}^{i}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3}$$

If $\lambda = 0$, we go back to the Linear regression defined above. The Ridge regression improves the stability of the regression by incurring additional costs for a model that has large coefficients.

## 2.3 K nearest neighbours regressor

K nearest neighbours is an algorithm that stores all available cases and predicts the numerical target based on a similarity measure such as euclidean distance [2]. Even though K-NN is easy to implement and can handle non-linearities, it is very slow to predict.

## 2.4 Random forest regressor

Random forest [3] is a Supervised Learning algorithm. It is also called a bagging technique since all the trees are run in parallel (each tree comes from a different bootstrap sample of the training data set). It aggregates all the decision trees, hence it is considered as a meta estimator. For a regression problem, a prediction is the average prediction across the decision trees.
Random forest has many advantages, for instance, it runs efficiently on large databases and it estimates missing data effectively. However, the algorithm tends to overfit some data sets with regression tasks and tends to be biased in favor of attributes with more levels when the data set contains categorical variables with different number of levels.

## 2.5 Gradient boosting regressor

Gradient boosting also relies on decision tree [4]. The term boosting refers to the aggregation of simple models. These are called weak learners and are added one at a time, while keeping existing trees in the model unchanged. An increasing number of simple models are combined, making the final model a good predictor. To minimize the loss (as for linear regression), the algorithm uses gradient descent. At each step a residual is computed, and is added to the input of the existing model, pushing the model towards the correct target.
This method is good to handle missing data, but also tends be more accurate than Linear regression, and to have a minimal data preprocessing.

# 3 Results

We used sklearn environment for all the models. To conduct the linear regressions, we used the "linear_model" package. The package "ensemble" was necessary for Random forest and the Gradient boost.
For each model, we fitted the data contained in "augmented" learn and used the model derived from it to predict Emolument based on data from "augmented" test.
For the random forest, we defined 60 number of trees, the maximum depth of the tree at 25, the minimum number of samples required to split an internal node at 20, the number of jobs to run in parallel at 2 and the number of features to consider when looking for the best split at 30. These numbers are based on predicting salary analysis ([5]), and are useful to avoid overfitting.
For gradient boosting, we activated the option of the loss function to optimize and the option to control the verbosity when fitting and predicting. Also, we reduced the maximum depth of the tree to 5. Regarding Knn started to find the number of nearest neighbour that allowed to get the lowest mean squared error. Then, we used this number to compute the model.

For each model, we used the 5-fold cross validation score as a resampling technique on the learning data and on the target value (emolument). We also computed the $R^2$ for each regression to have another measure. For random forest, we could compute the out of bag score.

| Method | Cross-Val | R2 |
|---|---|---|
| K-NN | 0.5047 | 0.5903 |
| Linear regression | 0.6826 | 0.6977 |
| Ridge regression | 0.6828 | 0.6977 |
| Random forest regressor | 0.7009 | 0.8438 |
| Gradient boosting regressor | 0.7155 | 0.7883 |

Table 1: Evaluation of the methods

Both linear regression and ridge regression are very easy to implement but do not give the best results. Based on the cross validation score, the best model is Gradient boosting. However relying on the $R^2$, the best model is the Random forest. The $R^2$ is neither too low, nor too high, suggesting that there is no overfitting/underfitting. With cross validation score of 0.7155, one can expect 71% accuracy on the test set which is a relatively good result.

# References


[1] URL: https://realpython.com/linear-regression-in-python/.

[2] URL: https://medium.com/analytics-vidhya/k-neighbors-regression-analysis-in-python-61532d56d8e4.

[3] URL: https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f.

[4] URL: https://blog.paperspace.com/implementing-gradient-boosting-regression-python.

[5] URL: https://github.com/rajpurohitpooja/Salary_Prediction_Portfolio/blob/master/Salary_Prediction.ipynb.