**Human Resources Analysis**

**Project description:** project that provides an ability to predict which employees will leave the company.

**Environment & tools:**

* Python (3.6.2);
* Pandas (0.23.4);
* Numpy (1.15.1);
* Scikit – learn (0.19.2);
* MatPlotLib (2.2.3);
* Scipy (1.1.0);
* Seaborn (0.9.0).

**Data:** dataset is simulated (<https://www.kaggle.com/ludobenistant/hr-analytics/downloads/HR_comma_sep.csv>)

 Fields in the dataset include:

* Satisfaction Level;
* Last evaluation;
* Number of projects;
* Average monthly hours;
* Time spent at the company;
* Whether they have had a work accident;
* Whether they have had a promotion in the last 5 years;
* Departments (column sales);
* Salary;
* Whether the employee has left.

**Type of task**: task of binary classification.

**Model:** model has two parts: training process and forecasting process.

Oracle Trainer Class implements training process.

Oracle Class implements forecasting process.

**Classifier:** Random Forest Classifier.

**Description of the model.**

Oracle Trainer Class.

Input data: path to file in CSV format.

Main method: train( ). It does not need the input data.

This method:

* Reads data from CSV file and convert it to Data Frame;
* Deletes duplicates and reset index in the Data Frame;
* Converts two categorical columns (sales, salary) into indicator columns (encodes labels with value between 0 and n\_classes-1);
* Splits Data Frame into dataframe (contains only features) and labels for this data;
* Splits dataframe (contains only features) and labels in train/test sets;
* Creates the classifier and fits it using training data and labels;
* Calculates accuracy score of classifier on the given test data and labels;
* Saves:

1. dictionary, which contains classifier and its accuracy score,
2. model, which encodes “sales” column,
3. model, which encodes “salary” column.

Output data: dictionary, which contains classifier and its accuracy score.

Oracle Class.

Input data: dictionary of answers on survey.

Dictionary should include the follow keys (only in such order):

* 'satisfaction\_level',
* 'last\_evaluation',
* 'number\_project',
* 'average\_montly\_hours',
* 'time\_spend\_company',
* 'Work\_accident',
* 'promotion\_last\_5years',
* 'sales',
* 'salary'.

Example:

answers\_dict = {**'satisfaction\_level'**: 0.2,  
 **'last\_evaluation'**: 0,  
 **'number\_project'**: 2,  
 **'average\_montly\_hours'**: 300,  
 **'time\_spend\_company'**: 5,  
 **'Work\_accident'**: 0,  
 **'promotion\_last\_5years'**: 0,  
 **'sales'**: **"IT"**,  
 **'salary'**: **"low"**}

Main method: predict( ). It does not need the input data.

This method:

* Transforms dictionary of answers into data frame.
* Uploads:

1. dictionary, which contains classifier and its accuracy score,
2. model, which encodes “sales” column,
3. model, which encodes “salary” column;

* Encodes values of “sales” and “salary”;
* Predicts the value (0 - stay, 1 - leave) according to the answers on survey.

Output data: number (0/1), which represents the result of the model and the confidence of model (float number, %) in its decision.