CMP4336 Introduction to Data Mining Course Project

Spring 2024

# Introduction

In our project we will generate a Wage Prediction Model and our target variable will be “SALARY” using Jupyter, Anaconda & Python.

Before getting into our data, we will first show and explain the environments and the language we have chosen to conduct this project.

**What is data mining?**

Data mining is a highly effective method for extracting key insights, meaningful patterns, and valuable knowledge from extensive datasets. A range of techniques such as statistical analysis, artificial intelligence, and machine learning are used to uncover hidden patterns, relationships, and trends within data. The ultimate objective of data mining is to provide decision-makers with valuable information that can be applied across various industries, including business, finance, healthcare, and science.



1. **Anaconda:**

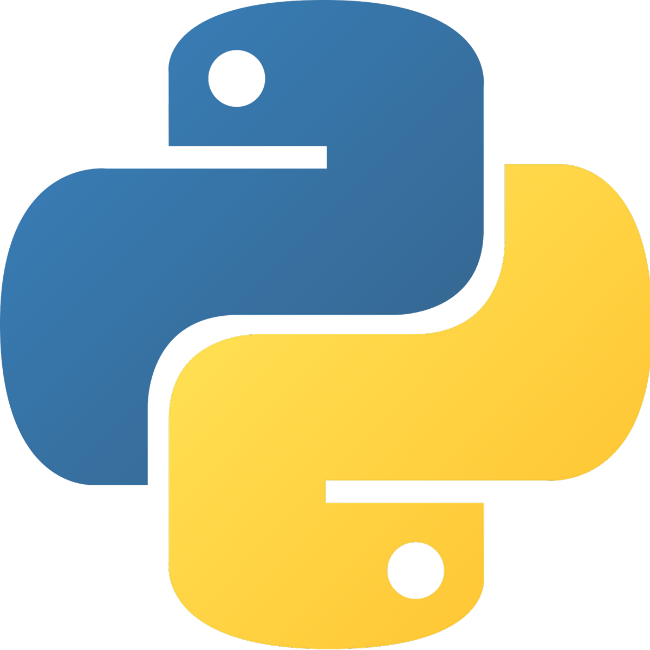
Anaconda is a versatile platform that is used for data mining. It provides a comprehensive package management system and creates an environment for programming languages like Python and R. Anaconda comes with a wide range of pre-installed data science tools and libraries, which makes it easy to set up and manage environments for data mining projects. With Anaconda, the process of installing various data mining tools is simplified. It also ensures compatibility between different libraries, allowing data scientists to focus more on the analysis and less on the setup and configuration.



1. **Jupyter Notebook:**

Jupyter Notebook is a remarkably adaptable instrument for data mining. It offers an interactive setting that enables users to explore, manipulate, and visualize data. Additionally, it supports experimentation, prototyping, and documentation, and seamlessly integrates with popular data science libraries like pandas and Scikit-learn. As a result of its collaborative features and interactive nature, data scientists frequently rely on Jupyter Notebook for a variety of data mining tasks.

1. **Python:**



Python is a remarkably versatile programming language renowned for its extensive use in data mining applications. It provides comprehensive support through libraries such as pandas has a gentle learning curve, a large and active community, and is both scalable and interoperable. Its flexibility and adaptability make it perfect for a broad spectrum of data mining tasks, from fundamental exploration to sophisticated analytics.

# Dataset Description

We have a firm’s recruitment interviews for positions of :

1) Data Scientist

2) Senior Data Scientist

3) Architect

In the last 1 year, it contains the candidates’ requested salary in dollars (annual net).

A screenshot of a computer

Description automatically generated

Information about the candidates interviewed for the position is taken from LinkedIn combo boxes, attached resume information, and information recorded in the interviews held by the HR Department or Head Hunters.

To start, we first check whether our requirements are met or not in Jupyter ( the installations of the tools…)

**Then we upload our data set to Jupyter.**

df = pd.read\_excel('datam.XLSX')

with pd.option\_context('display.max\_rows', None):

display(df)

df.head()

**After successfully displaying the data set, we will look at “Empty Registration Amounts”**

display(df.isna().sum())

Salary 0

Role 0

Total\_Exp 0

Otg\_Job\_Exp 14

Other\_Exp 14

Master\_Degree 10

Club\_Membership 38

Preferred\_Job\_Location 44

English\_Level 0

Other\_Languages 20

University\_Degree 10

Hobbies 165

License\_Certificate 171

Tec\_Talents 0

dtype: int64

**Then we look at the data types in the set.**

df.dtypes

Salary int64

Role object

Total\_Exp float64

Otg\_Job\_Exp float64

Other\_Exp float64

Master\_Degree object

Club\_Membership object

Preferred\_Job\_Location object

English\_Level object

Other\_Languages object

University\_Degree float64

Hobbies object

License\_Certificate object

Tec\_Talents int64

dtype: object

# Methods

**We do a Categorical variable analysis and a Numerical variable analysis.**

We looked at missing values and we did not want to add the data according to the median or mean. Since we do not have a lot of data in our hands, we wanted to look at and test the variables one by one.

**We compare with the variables from the Numerical variable analysis.**

**For Otg\_Job\_Exp - Experience in present job,** there is a big difference between mean and median, and when you look at the distribution chart, although the average is 3.12, the data is more intense at values ​​1 and 2.

For this reason, we will fill the empty data with the median in this variable.

**For Other\_Exp,** there is not much difference between mean and median for this variable. Looking at the distribution graph, it shows that there is more data between 0 and 1. Since variables such as work experience may be directly related to salary, we will fill in the empty data here with the smaller value of 2.5, that is, the median

**For Master\_Degree,** the yes answer is quite low, and we thought that if a candidate had a master's degree, he would add this for this reason, we will fill the empty data with the answer No.

**For Club\_Membership,** is the same logic as master\_degree.

**For Preferred\_Job\_Location,** In this data, we will apply the fill empty data with the most repeated value method, which is the most commonly used technique in categorical variables.

**For Other\_Languages,** We will create a new categorical value called "None" for missing data.

**For University\_Degree,** it is hard to comment on this variable because it isinterpretation, both as a continuous variable and as a success criterion, may differ from company to company. To penalize empty data we filled with median.

**For Hobbies and License\_Certificate** we will fill empty data with NO.

**We have filled in the empty data. Now we will go into distribution analysis.**

Writing code for each variable individually is unnecessary and a waste of time, so we will define a group for numeric variables and categorical variables.

numeric\_columns = df.select\_dtypes(include=['int64', 'float64']).columns

categorical\_columns = df.select\_dtypes(include=['object']).columns

# Experimental Results

**LINEARITY**

We conduct RAINBOW TEST

H0: There is a linear relationship between the dependent variable and the independent variable.

H1: None.

The p-value is above 0.05 in all independent variables except Otg\_Job\_Exp;

To reject H0, it had to be less than 0.05

Therefore, "There is a linear relationship between the dependent variable and the independent variable."<br>

We cannot reject his hypothesis.

According to both the scatter plot and rainbow test, we can say that our data is **linear.**

**INDEPENDENCE OF ERRORS**

RESIDUAL PLOT ANALYSIS & DURBIN WATSON TEST

Values ​​close to 2: Error terms are independent and there is no autocorrelation.

Values ​​close to 0: Positive autocorrelation.

Values ​​close to 4: Negative autocorrelation.

In our value, (there is also the possibility of negative autocorrelation very close to 2), we can say that the error terms in our data are independent.

There is no autocorrelation.

**HOMOSCEDASTICITY**

BREUSCH-PAGAN TEST

Both tests showed p-values ​​lower than the reference value of 0.05. For this reason, homogeneity of variance is not present in our data.

WHITE TEST

H0: The variance of the error terms is constant in all observations

H1: Error terms vary across observations.

Our p-value is lower than the reference value of 0.05. H0 is rejected. There is no homogeneous distribution in the error terms. Homogeneous distribution in error terms is a desired condition.

**NORMALITY OF ERRORS**

Q-Q PLOT ANALYSIS

SHAPIRO-WILK TEST

H0: The data comes from a normally distributed population.

H1: The data does not come from a normally distributed population. The p-value is much smaller than the reference value of 0.05.

H0 is rejected. The distribution of error terms does not come from a normally distributed population.

KOLMOGOROV-SMIRNOV TEST

H0: The data comes from a normally distributed population.

H1: The data does not come from a normally distributed population.

The p-value is much smaller than the reference value of 0.05.

H0 is rejected. The distribution of error terms does not come from a normally distributed population.

**MULTICOLLINEARITY**

CORRELATION MATRIX

Although + 80% results indicate a high appearance, considering the 80% threshold in a model with such a small number of observation points and explanatory variables, low degrees of freedom would be a reason for overfitting.

Experience variables are all variables that can change with each other. So we will make a small analysis,

**Are they available in the salary source based on the person's previous work experience or their total experience?**

**VARIANCE INFLATION FACTORS (VIF)**

vif < 5 low multicollinearity

5 <= vif < 10 medium multicollinearity

vif >= 10 high multicollinearity

Infinite values ​​indicate the VIF problem, as we expected. All of these values ​​are affected by each other.

**IQR ANALYSIS**

* IQR CAPING
* Z-SCALE IN NUMERICAL DESCRIPTIVE VARIABLES
* ASSIGNING DUMMY VARIABLES TO CATEGORICAL VARIABLES

A table with numbers and letters

Description automatically generated

**CORRELATION BETWEEN INDEPENDENT VARIABLES**

Apart from dummy variables, only Total\_Exp and Role\_Data\_Scientist variables correlate with the threshold value of 0.8.

Define correlation thresholds

**CORRELATION BETWEEN TARGET AND INDEPENDENT VARIABLES**

The most important result we see here is that Total\_Exp and Other\_Exp variables have a higher correlation than Otg\_Job\_Exp. We will take this inference into consideration when creating data sets.

**ESTABLISHING THE LINEAR REGRESSION MODEL**

**Equation of the model:**

Salary = 53434.36 + -2838.58 \* Other\_Languages\_German + 3923.01 \* Role\_Architect + 7916.14 \* Total\_Exp + -6705.53 \* Role\_Data\_Scientist + 3981.50 \* Other\_Exp + 2782.52 \* Role\_Senior\_Data\_Scientist + -2922.17 \* English\_Level\_Limited\_Working

R^2 has decreased but it is hard to try it one by one. So we will look at the top 5 combinations out of 100 iterations and print them.

The correlation between our error terms was high. For this reason, we add grid search to our model.

Cross-Validation R-squared Scores: [0.82201671 0.89648712 0.94921218 0.95983005 0.95883253]

Mean Cross-Validation R-squared: 0.9172757156245621

Coefficients of the Best Model:

Total\_Exp: 13379.014507262204

Other\_Exp: 3233.088130722555

Master\_Degree\_Yes: 897.7849389443167

Preferred\_Job\_Location\_Hybrid: -3024.19490952651

English\_Level\_Limited\_Working: -1795.1366508562696

# Conclusions

R-squared of the Best Model (Test Set): 0.9020307290321163

**Initialize a list to store results for linear regression and random forest**

Created a directory to save decision tree images

We have successfully generated a Wage Prediction Model with our target variable as SALARY.

Additionally, we have seen that information that we memorize is not always the right way to go along with models. Deploying this model also showed us that coding is not the hard part but analysing and manipulating data in a way that will benefit us is. This project created a new perspective for us regarding data and its importance in life. It was exciting to use theoretical knowledge in practice.