VNUHCM – UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

★★★

REPORT PROJECT 03: LINEAR REGRESSION

Course name: Applied Mathematics and Statistics
Class: 21CLC10
Students:

Nguyễn Thị Minh Minh 21127528

Instructors:

Ta. Phan Thị Phương Uyên
Ta. Vũ Quốc Hoàng
Ta. Nguyễn Văn Quang Huy
Ta. Ngô Đình Hy

Ho Chi Minh City – August 19, 2023

TABLE OF CONTENTS

I. DETAILED INFORMATION

1. Personal information:

ID	Full name	Gmail
21127528	Nguyễn Thị Minh Minh	ntmminh21@clc.fitus.edu.vn

2. Plan assignment & Self assessment:

Task	Assessment
Task 1a: Use total 11 first features Gender, 10percentage, 12percentage, CollegeTier, Degree, colledgeGPA, ColledgeCityTier, English, Logical, Quant, Domain.	100%
Task 1b: Build model using only 1 characteristic feature, choosing from: conscientiousness, agreeableness, extraversion, neuroticism, openess_to_experience.	100%
Task 1c: Build model using only 1 feature, choosing from English, Logical, Quant.	100%
Task 1d: Student build the model, and find the model with best result.	100%

II. LIBRARY IN USE:

1. Library pandas:

This library will do the duty of read data from csv file to form data set for training and testing.

2. Library numpy:

This library will help perform calculation on data matrix.

3. Library matplotlib:

The duty of this library is to display the image showing the data performance.

4. Library seaborn:

I use this library to explore data with the correlation heatmap.

5. Library sklearn:

a. sklearn.model_selection import KFold:

- KFold cross-validator provides us train indices and test indices to split the train data set into k consecutive folds. [1]

class sklearn.model_selection.KFold(n_split=5, *, shuffle=False, random_state=None**)**

- For example:

```
X_train = [1, 2, 3, 4, 5, 6]
subsamples = KFold(n_split=3, shuffle=False).split(X_train)
for train_index, test_index in subsamples:
        print("Train indices": train_index);
        print("Test indices:" test_index);
```

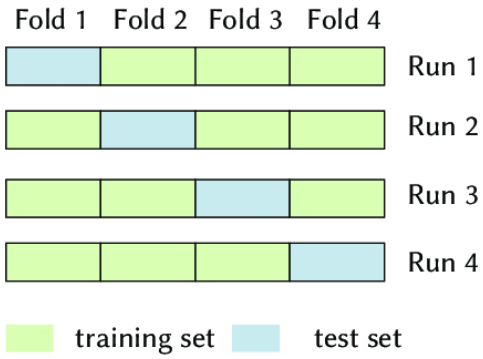
- The output is:

```
Train indices: [2 3 4 5]
Test indices: [0 1]

Train indices: [0 1 4 5]
Test indices: [2 3]

Train indices: [0 1 2 3]
Test indices: [4 5]
```

- The code above is describing the workflow of KFold. As the function will return the indices of train and test. Each loop will return the current set as testing set and the remaining will become the training set.



b. sklearn.linear_model import LinearRegression:

- The library is used to fits a linear model. Besides, I have also implemented a version of LinearRegression writen by my self. The workflow of the two classes is quite equivalent to each other. In this project, at the question for Task 1b, 1c and 1d; I call LinearRegression to fit my data. [2]
- Main used function
    - fit(X_train, y_train) with parameters fit_intercept: bool, default=True.
        - If fit_intercept is set to False, no intercept will be used in calculation. This stage is the process of inserting the column 1 in the training matrix.
    - predict(X_test): used to predict value of test data set.
    - Used attributes:
        - coef_ : Estimated coefficients for the linear regression problem.

- intercept_ : This is the independent term in the linear model.

c. sklearn.metrics import mean_absolute_error:

sklearn.metrics.mean_absolute_error(y_true, y_pred)

- MAE is mathematical measure to measure the average magnitude of errors between the actual and predicted values. [3]
- The mean absolute error is used with the formula:

$$\text{MAE} = \frac{1}{n}\, sum_{i=1}^{n} |y_i - \hat{y}_i|$$

d. sklearn.feature_selection import mutual_info_regression:

- The mutual_info_regression is a part scikit-learn, designed to select feature for regression task with the purpose of predicting a continuous target variable. Overall, mutual information measures will help us have insight into correlation on information one variable giving for another variable. Therefore, this library is really helpful for feature selection that strongly related to the target. [4]

sklearn.feature_selection.mutual_info_regression(X,y)

- **X:** The feature matrix, each row will present a sample and each column will present a feature.
- **y:** The target variable. In this case, the target variable is salary.

e. sklearn.feature_selection import SelectPercentile:

- This class enables us to select a specific percentage of the most relevant features from the dataset. [4]
- The class signature:

sklearn.feature_selection.SelectPercentile(mutual_info_regression, percentile=n)

f. Sklearn.model_selection import train_test_plit

- It is used for splitting a dataset into two subsets: a training set and a testing set. [13]

sklearn.feature_selection.train_test_split (*arrays, test_size=None, train_size=None, random_state=None, shuffle=True, stratify=None)

- *arrays: One or more arrays (or sequences) of data.
- test_size: Size of the test.
- train_size: Size of the train.
- random_state: An optional random seed for reproducibility.
- shuffle: An optional random seed before splitting.

- stratify: it's used for stratified sampling based on the class labels.

6. **Library statsmodels.api**

- In this library, I have used OLS class to fit data. Overall, the workflow is the same as in library **sklearn** but I decide to it because it give me detailed information to get **pvalues**, **tvalues** to find feature in task 1d.
  - **OLS**(X_train, y_train).fit()
  - **.params:** to get the coefficient.
  - **.tvalues:** to get t-statistic value.
  - **.pvalues**: to get p-value.

III. FUNCTIONS:

1. Class OlinearRegression:

a. Function def fit: [5]

- Input: X_train, y_train
  - X_train: the matrix of training data.
  - y_train: the target value of training data.
- Output: self
  - The class object is updated with the linear regression parameters.
- Description: This function computes the linear regression using pseudo-inverse method. The `w` is the result of regression coefficients.
  - The used formula: $(X_{\text{train}}^T X_{\text{train}})^{-1} X_{\text{train}}^T \cdot y_{\text{train}}$

b. Function def getParams:

- Input: object (no input parameters).
- Output: object.w
  - Vector containing the regression coefficients, and the result will be rounded to 3 decimal places.
- Description: This function will return the regression coefficients after calculating the `fit` function, and the result is rounded to 3 decimal places.

c. Function predict:

- Input: X. This is the feature matrix of testing data set.
- Output: array containing the predicted values corresponding to each sample in `X`.
- Description: The function calculates the predicted values by let the matrix X multiply with each regression coefficient in `self.w`, then summing the results.

2. Function def preprocess:

- Input: X_train, y_train
  - X_train: a 2D array representing the feature matrix of training data.
  - y_train: a 1D array representing the target values of training data.
- Output: X, y_train
  - X: is a modified feature matrix from X_train by adding column of ones at the first column.
  - y_train: the original `y_train`.

- Description: The function preprocesses the training data by inserting the column of one to the matrix X_train. In Linear Regression, this column is used for the bias term in linear regression models. [5]

3. **Function def mae:**
   - Input: y, y_pred
     - y: a 1D array of the actual target values from data set.
     - y_pred: 1D array of the predicted values that have been calculated from the function `predict` above.
   - Output: mae result calculating from y and y_pred.
   - Description: By using numpy, the function calculates MAE between `y` and `y_pred`. By computing the average of the absolute differences between `y` and `y_pred` via the formula.

$$\text{MAE} = sum_{i=1}^{n} |y_i - \hat{y_i}|$$

   - Lower MAE values will indicate better predictive accuracy.
4. **Function def calculate_feature_mae:**
   - Input: X_train, y_train, k_fold, feature_dict
     - X_train: trainning data set.
     - y_train: target value of data set.
     - k_fold: the number of folds to be used in cross-validation.
     - feature_dict: this is a dictionary I designed to hold the features with their corresponding values.
   - Output: Dataframe from the updated Feature-Mae dictionary.
   - Description: This is the function I will use for task 1b and 1c. I use k-fold cross-validation to calculate MAE.
     - KFold function first will shuffle all data once then return the train indices and test indices.
     - For each feature in the given dictionary, the function will iterate through k folds with train indices and test indices return from KFold function and train the model with Linear Regression. After trainning, it continue calculating the MAE on the corresponding test fold using `mean_absolute_error` from sklearn, each feature will have their accumulated MAE through n iterations (n is the number k-folds). Finally, the accumulated MAE values are devided by `k_fold` to gain the average MAE for each features; MAE will be rounded to 3 decimal places.
5. **Function def correlation:**
   - Input: dataset, threshold.
     - dataset: the data Dataframe read by pandas for correlation analysis.
     - threshold: a value – a threshold to choose features whose the correlation values is assessed as higher.
   - Output: a set of chosen feature names possessing high correlation values over the threshold.

- Description: Firstly, I create a set() variable named `col_corr` to save the return values. Secondly, I will call the function `.corr()` from the library `seaborn`. The seaborn function will return a table of correlation values with the axis 1 with 23 features, axis 0 with 23 features. After that, I will use nested loop to traverse through each position to check whether the correlation is higher than the threshold or not. If the values is higher than the threshold, the feature will be chosen and add to `col_corr`.

6. Function def average_mae:

- Input: X_train, y_train, k_fold
  - X_train: trainning data set.
  - Y_train: taget values of training data.
  - k_fold: integer type representing the number of folds in cross-validation.
- Output: return the average MAE after using Linear Regression model.
- Description:
  - The function first shuffle and splits the training data into k folds.
  - Now I will use a loop to traverse through these folds to build the model on the trainning set with LinearRegression and calculate the accumulated MAE with `mean_absolute_error` on the corresponding test subset. The MAE values finally, will be divided by `k_fold` to compute the average MAE on all k folds.
  - The function will return the average MAE that is rounded to 3 decimal places.

IV. EVALUATION ON THE RESULT:

1. Task 1a:

a. Result:

- Model:

Salary =	49248.09 – 23183.33 **Gender** + 702.767 **10percentage** + 1259.019 **12percentage** – 99570.608 **CollegeTier** + 18369.962 **Degree** + 1297.532 **collegeGPA** – 8836.727 **CollegeCityTier** + 141.76 **English** + 145.742 **Logical** + 114.643 **Quant** + 34955.75 **Domain.**

- MAE on test train:

MAE = 105052.53

b. Evaluation:

- From the built model, it is clearly to see that the Salary in India is under high influence of academic and degree factors. The biggest impact on Salary is clearly observed that **Domain** and **Degree** since they own the two biggest coefficients, 34955.75 and 18369.962 respectively. This is also easy to understand because in such a comparative society, the possession of degree is one of the most important thing to gain a position in society. **Degree** could be used as the soundest proof to show with the HR that you are really a trained skillful candidate, thus gaining the

advantage of dealing a higher salary range. Concerned about **Domain**, this is the point showing the process what you have learned in the university and displaying part of your capabilities in the sector.

- Following after is the feature of **10percentage**, **12percentage** and **collegeGPA**. These features belong to the academic group, also showing your effort in school years. Hence, based on the model, the mentioned group has equivalent ratio with the salary. Meaning that, when record on school years is higher, the salary also varies in the up direction. The same situation also happens with **English**, **Logical** and **Quant** group, these features have positive coefficients.
- However, there are three negative coefficients existed in the model with **Gender**, **CollegeTier** and **CollegeCityTier**. In perspective of **CollegeTier** and **CollegeCityTier**, it can be understood that universities with lower **CollegeTier** index is top schools with higher education standard. For example, school top 1, of course is better than school top 100. This way of explanation is also similar when applying to **CollegeCityTier**; higher index of **CollegeCityTier** means that students locate in the countryside and vice versa. So, we can conclude that the higher value of **CollegeCityTier** and **CollegeTier** leads to the drop in salary.
- About **Gender**, in reality, this is a harsh truth that in India, the situation of disparity in gender is still a matter problem. And this fact is also reflected in the above model in the problem of salary. As with inverse ratio, when gender equals to 1 – male implication, the salary decreases less than gender with the value of 2 – female implication.

2. **Task 1b:**
   a. Result:

	Feature	MAE
0	neuroticism	123453.725
1	agreeableness	123592.732
2	openess_to_experience	123939.743
3	extraversion	124051.763
4	conscientiousness	124186.926

- In this task, there exist a little volatility as the feature `**neuroticism**` and `**agreeableness**` could swap the position to become the best feature. However, the probability of `**agreeableness**` being the best feature is quite low in comparison with feature `**neuroticism**`. Therefore, I decide to choose `**neuroticism**` being the best feature.

- The model of `**neuroticism**` trained on the total train set:

  Salary = -16021.494 **neuroticism** + 304647.553

- The MAE value on the test set:

  MAE = 119361.917

b. Evaluation:

- After using the technique assessing the model with k-fold cross validation, 5 models under consideration are **conscientiousness**, **agreeableness**, **extraversion**, **neuroticism**, **openess_to_experience**, belonging to the characteristic group. The model of **neuroticism** is chosen to be the best feature producing the best model. In other words, **neuroticism** has the best salary predictability.
- This can be explained as follows: **neuroticism** shows the personal character; when the value of this feature increases, the **salary** tends to fall.
- With my knowledge, I understand that person with high value of **neuroticism** is usually stick to the problem of anxiety, emotional instability, stress sensitity, overthinking and physical symptoms like headaches, muscle tension as well. Acknowledegeablly, the productivity of those people, of course can be affected severely, contributing to the lower salary range.
- A paper research is also conducted on this case[6], it's also written that '*Neurotiscim had negative associations with satisfaction variables. Pay and job satisfaction were positively correlated'* . The paper highlighted the intricate relationship between personality traits and salary levels, enforcing my point of view presented above.
- In conclusion, high levels of **neuroticism** could potentially influence the payment, as the negative impact on the job performance, engagement and advanced opportunities within the career. Realizing its impact, I think all of us should practice ourselves to create a positive attitude toward the job we are doing.

3. **Task 1c:**

a. Result:

	Feature	**MAE**
0	Quant	117249.104
1	Logical	120144.660
2	English	120776.499

- The chosen best model is **Quant.**
- The model of **`Quant`** trained on the total train set.

  Salary = 368.852 **Quant** + 117759.729
- The MAE value on the test set:

  MAE = 108814.06

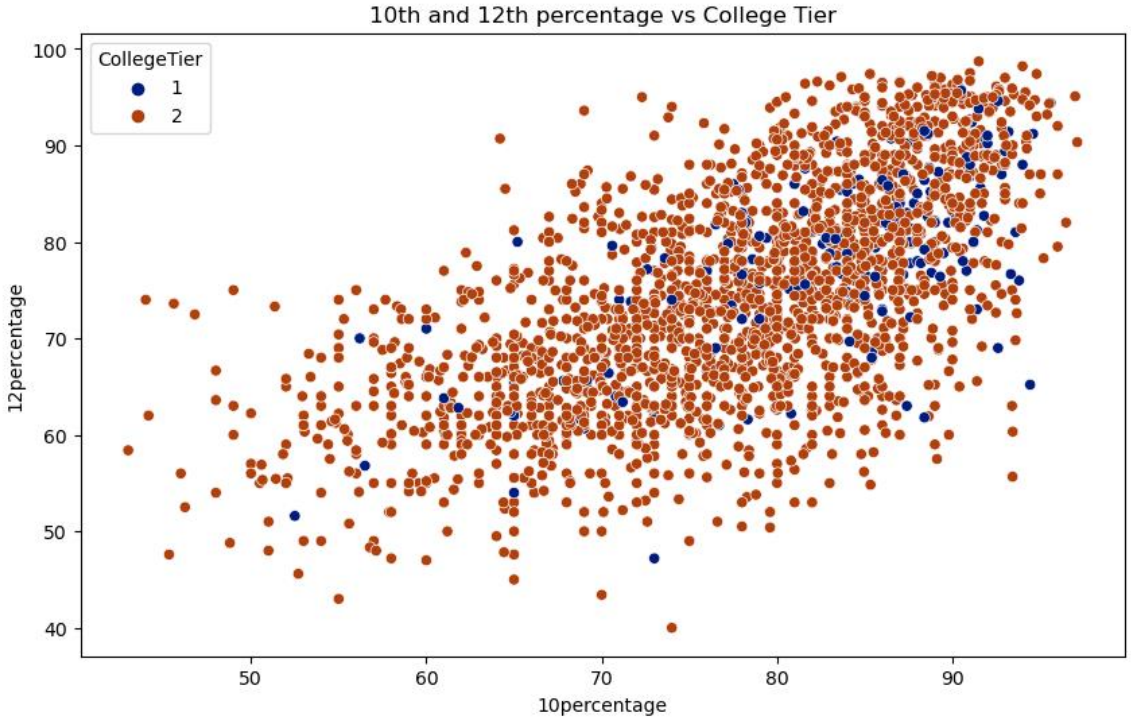
b. Evaluation:

- The result table when applying k-fold cross validation show that among three features in terms of cognitive skills, **Quant** is the factor achieving the highest impact on salary with MAE = 117249.104.
- Besides **Logical** and **English** skill, the term **Quant** or quantitative skill, for some people, is a new concept. To define it more clearly, a quantitative skill is any skill that involves using or manipulating numbers. It is the ability to reason using numbers [7]. Several important aspects to quantitative ability is listed: data

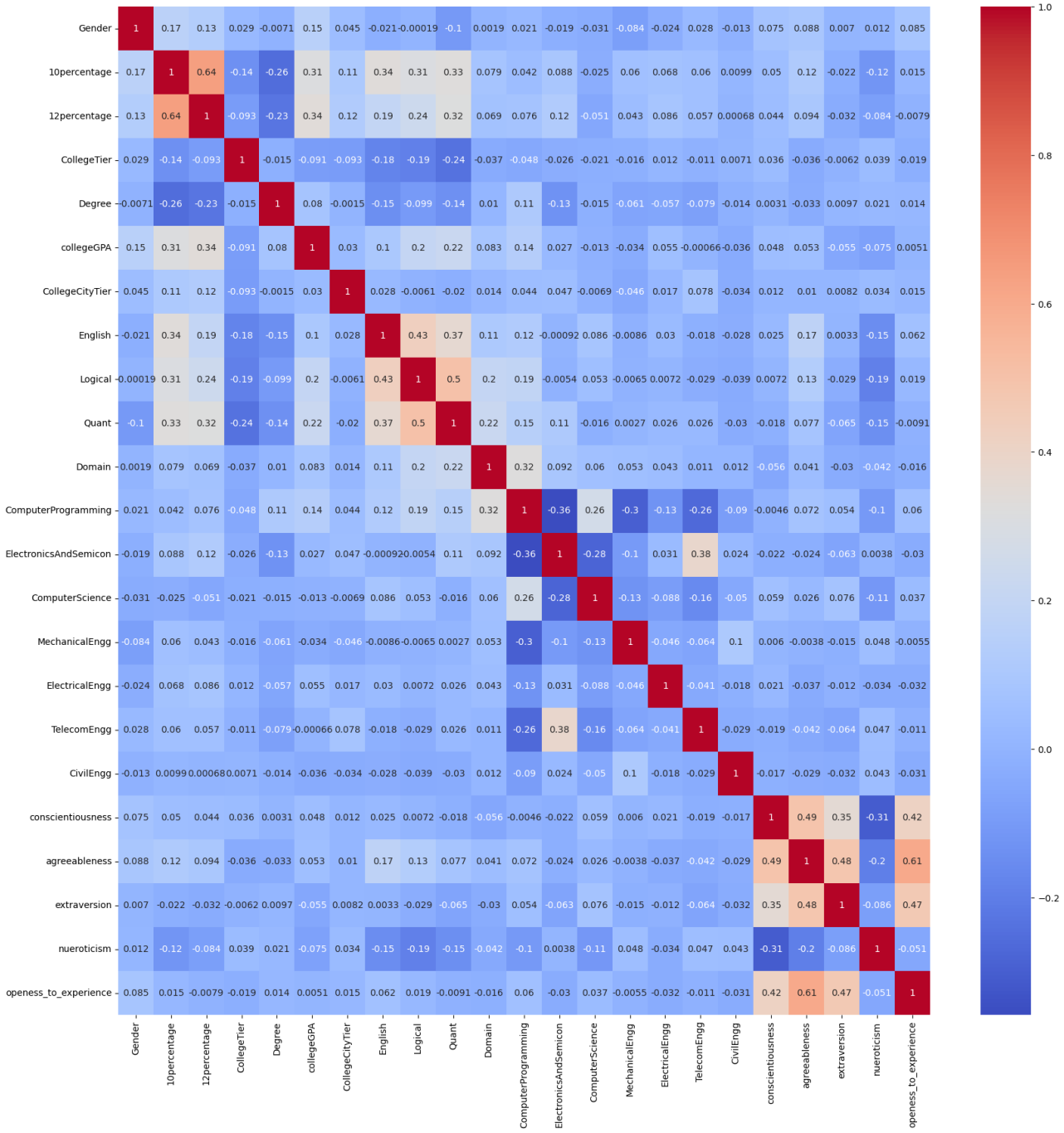
collection, numerical representation, data analysis, interpreting results, prediction and management.

- The model displays that when the score of **Quant** increases, the salary also has the same pattern. This is so practical and understandable as people with a calculating brain could make a considerable contribution to the quality and productity of work. As a result, high salary could attract talented quantitative human resources to the organization.
- Besides, although **Logical** and **English** has lower effect on salary than **Quant**, personally I still take the two factors into list of crucial skills serving our career path.

4. **Task 1d:**
    a. Finding model 1:
    - [8][9] The main idea I use on selecting features for the first model is using correlation coefficient. With the help of library pandas, I could measure the linear relationship among variables. By calculating the linear relationship among variables within a dataset, I could assess the dependency of variables on the another. If the two variables are highly correlated, we could make predictions one from the other.
    - The core meaning here is that if two variables that has a high correlated value over a threshold, we only need to choose one of them because from one variable can predict the another.
    - For example, when calculating, I observed that the correlation value between **10percentage** and **12percentage** are 0.65, pretty high. Then I visualize it with scatterplot from the seaborn. The two features via the figure is really the same, called *multicollinearity*. Hence, in my chosen feature, I only choose 1 of them to predict the salary.



- By using the function from library pandas **.corr()**, and the function **correlation** written, I set a threshold at 0.45 to choose out most correlated features on salary.
- The heatmap is drawn with library seaborn:

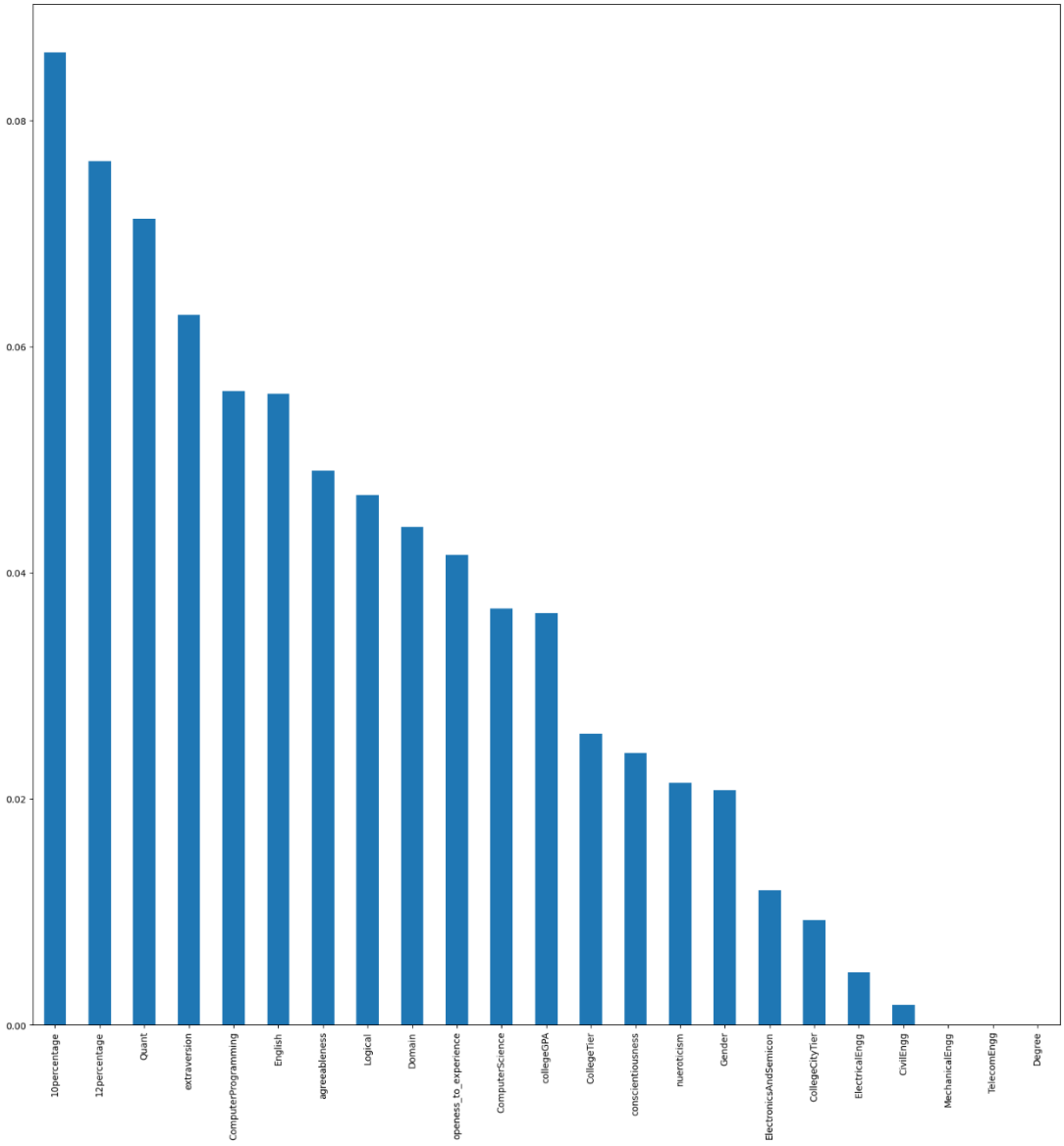


- From the heatmap, we observe that the dark zone is centered near the main diagonal with the value around 0.45. These zone, therefore, has the highest correlation within the dataset. And I decide to choose 0.45 as the threshold to pick out the features with greater 0.45 threshold values. Features include **12percentage, Quant, agreeableness, extraversion** and **openess_to_experience**.
- In conclusion, I found my model 1 is Linear Regression with 5 features:
  - 12percentage
  - Quant

- agreeableness
- extraversion
- openess_to_experience

b. Finding model 2:

- [10] [4] According to scikit learn, mutual information between two variables is a non-negative value, it will assess the dependency between the variables. If the value equals to 0, two variables are independent; else higher values mean higher dependency.
- In this model 2, I will measure the dependency index between each variable with the target value via the function **mutual_info_regression** from the library sklearn. The purpose is to evaluate the dependencies between the target and its features.
- By calling the function **mutual_info_regression** and using plot bar, I get the figure:

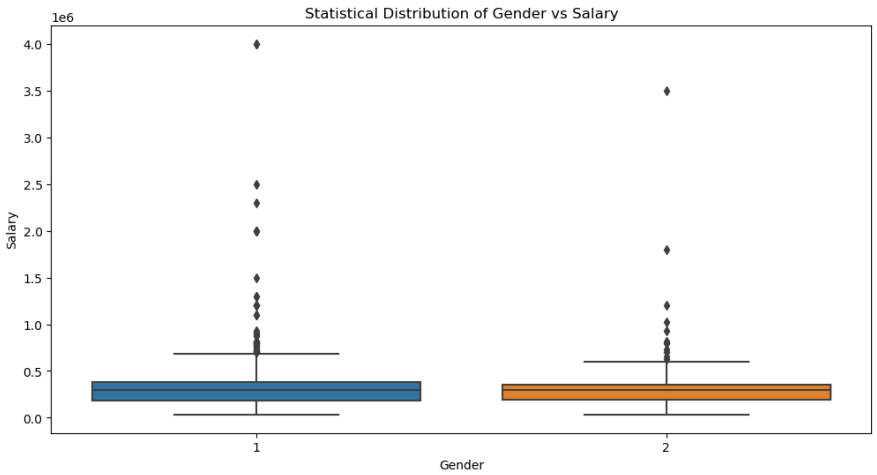


- The figure above is sorted from the highest dependency to the lowest dependency via the method **mutual information regression**.

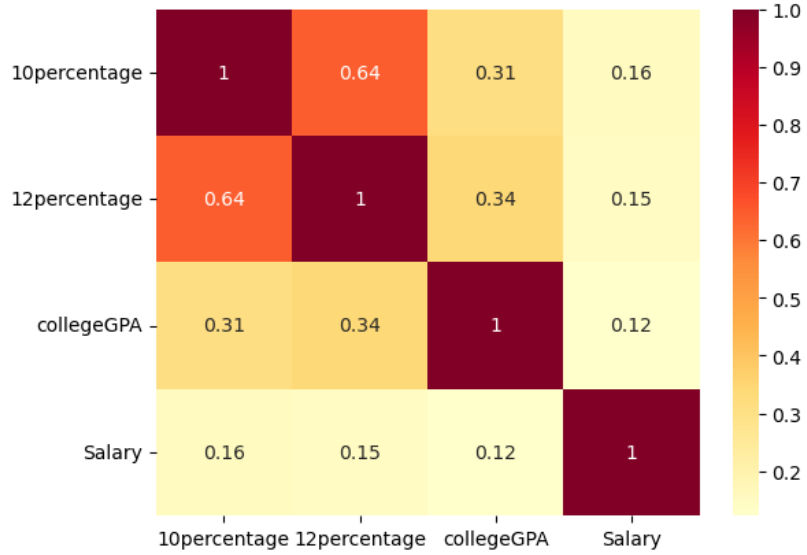
- Now, with the help of library **SelectPercentile** of sklearn, I choose top 20% features that have highest dependencies.
- The return features include: **10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, conscientiousness, extraversion, nueroticism** and **openess_to_experience.**
- In conclusion, the found model 2 is Linear Regression with 11 features:
  - 10percentage
  - 12percentage
  - collegeGPA
  - English
  - Logical
  - Quant
  - Domain
  - conscientiousness
  - extraversion
  - nueroticism
  - openess_to_experience

c. Finding model 3:

- [11] I based on method referred in this reference to find my model. Before I started to choosing feature for data fitting, I will devide the total data set into 4 aspects:
  - Academic variables: 10percentage, 12percentage and collegeGPA.
  - Cognitive skills: English, Logical and Quant.
  - Standardlized Test Scores: openess_to_experience, nueroticism, extraversion, aggreeable, conscientiousness and domain.
  - Specialization: ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, ElectricalEngg, TelecomEngg and CivilEngg.
- Now I will go through each aspect to choose variable in each group.
- But before that, I will make a comparison between Gender and Salary. By using statistical distribution of Gender and Salary, I have the followed figure. Observing the graph, it is clear that the salary range is quite equivalent despite the gender. Therefore, this feature will not be taken into consideration.



- Considering first group: **Academic Variables**
  - In this group, I will use correlation between salary and features in group academic variables.



  - Overall, the correlation between salary and other features is quite weak. However, correlation among features in group of Academic variable is quite good. Therefore, all forementioned 3 features are still significant contributors predicting the salary as changes in one academic-related feature could provide insight into how the other related feature might change, thus affecting the salary prediction.
  - So, I decide to choose all 3 features: **10percentage, 12percentage, collegeGPA**.
- Considering second group: **Cognitive Skills**
  - The main idea used for this group as well as group **Standardlized Test Scores** and **Specialization** is working with the value **p-values** and **t-statistics.** The concept of the two mentioned term is not strange with us, because this is concluded in the previous math – *Probability statistics.* Overall, I use them to assess the significance of the features in model.

- I will fit the model with those chosen model first. But in this case, I don't use LinearRegression from sklearn, I use OLS from the library **statsmodels.api** because I want to extract information like t-statistics (**.tvalues)** and p-values(**.pvalues)**.
- However, besides 3 features in group, I also add **Gender** to work as a control variable.
- So what is control variable? According to my reference [12], a control variable is a variable that is included in the analysis to get impact on the relationship among features. In this case, the main features are cognitive skills, and the outcome is salary, control variable is *Gender*. *Gender* helps isolate the result significance on the chosen features then better result is produced; else one of the features in cognitive group could be chosen as the control variables leading to less precise result.
- The table of value is as follows:

	t-statistics	p_values
English	4.087	0.000046
Logical	2.998	0.002753
Quant	5.589	0.000000
Gender	-1.434	0.151660

- Gender has **p_value** much greater than the other three features, so drop it.
- In terms of **Logical**, the **t_statistic** show that this feature seems to be insignificant when comparing with **English** and **Quant**. So I decide to drop the feature **Logical**.
- In conclusion, the chosen features in cognitive skills group is **English** and **Quant**.

- Considering the third group: **Standardlized Test Scores**
  - Applying the same method of the previous group, but with a little adjust, this time I will consider the two value **coefficent** and **t-statistics.** Using both values will ensure a more comprehensive understanding between the variables ad their significance.
  - The table of values is as follows:

	Coefficient	t-statistics
openeness_to_experience	-4615.652	-0.741
nueroticism	-17297.466	-3.485
extraversion	-207.679	-0.035
agreeableness	30066.940	4.32
conscientiousness	-24559.938	-4.277
Domain	50429.659	4.939
Gender	-24852.974	-2.187

  - From the above statistic feature with coefficients and t-statistics, I can see that clearly **Domain** and **agreeableness** possess highest coefficient and t-statistics. Therefore, the significance and practical impact on the salary of the two features would be the biggest in this group. In the

remain features, I still see there is a large gap between the **extraversion** and others, the coefficient and t-statistics of **extraversion** is bigger than others quiet much.

- In conclusion, **Domain, agreeableness** and **extraversion** are chosen.

- Considering the final group: **Specialization**

	Coefficient	**t-statistics**
ComputerProgramming	203.359	7.329
ElectronicsAndSemicon	64.87	1.823
ComputerScience	-152.040	-5.269
MechanicalEngg	162.049	3.033
ElectricalEngg	-90.009	-1.652
TelecomEngg	-61.321	-1.209
CivilEngg	170.129	1.17
Gender	-24391.277	-2.159

- Similar explanation with above, in this group, I choose 4 features: **ComputerProgramming**, **ElectronicsAndSemicon**, **MechanicalEngg**, **CivilEngg** which have highest coefficients and t-statistics.

d. Evaluation:

- Using k-cross validation, I got the dataframe with Model and their MAE values as follows:

	Model	**MAE**
0	MODEL3	113230.96
1	MODEL2	113937.489
2	MODEL1	116147.089

- After using the MAE test train, I can conclude that **MAE** of the Model3 is lowest so this is my best new model. However, there is a little notation is that the data is taken random so the number after different run could be not the same. Especially, Model3 and Model2 could swap to become the best feature. But the probability of Model2 is smaller through different test. So I decide to choose Model3 as the best feature.
- The new Linear Regression on Model 3:

Salary=692.034·**10percentage**+914.832·**12percentag**e+1319.187·**collegeGPA**+178.172·**English**+204.223·**Quant**+21464.018·**Domain**+95.965·**ComputerProgramming**+0.461·**ElectronicsAndSemicon**+111.833·**MechanicalEngg**+177.135·**CivilEngg**+7182.855·**agreeableness**−1709.676·**extraversion**−151323.732

- The MAE value of Model 3 on the test set:

MAE = 103764.375

- Overall, I am successful in building the new model because the value of MAE is smaller than all of the above. Because in model 3, I have choosen the model with

feeatures covering almost aspect in life. Therefore, the model performs better result.

V. REFERENCES:

[1]. KFold in sklearn.
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html
[2]. LinearRegression in sklearn.
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
[3]. Mean abosolute error in sklearn.
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html
[4]. Information gain. Mutual information gain.
https://github.com/krishnaik06/Complete-Feature-Selection/blob/master/
[5]. Lab 04.
[6]. Neuroticism affect on salary search paper.
https://www.researchgate.net/publication/365103971_Neuroticism_Materialism_Pay
[7]. Quantitative skill affect on salary.
https://www.uwb.edu/academic-support-programs/.
[8]. Feature selection techniques.
https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning
[9]. Feature selection techniques with heatmap correlation.
https://www.kaggle.com/code/nitinchoudhary012/engineering-graduate-salary-prediction?
[10]. Mutual information regression.
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html
[11]. Regression study with salary in India.
https://arrow.tudublin.ie/cgi/viewcontent.
[12]. Definition of control variables.
https://takelessons.com/blog/what-is-a-controlled-variable-in-science
[13]. Train split test in sklearn.
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html