**DOUGLAS COLLEGE – Winter 2025**

**CSIS 3290 – 050 – (Guided) Project 1**

**Your project will be considered to be a PLAGIARISM if:**

1. **The project is similar to the work submitted by other students within the same term, other terms or other sections;**
2. **The project is similar to any work found in the Kaggle or any other websites**

**Project resulted from plagiarism will receive a ZERO mark and the students associated with the work will be reported to the Dean office for academic dishonesty case**

## Project Description:

In this project you will be working on the prepared dataset from Lab2 that contains information of several tens of thousand rows of cars dataset. You will need to implement several linear regression models to predict the Car's price.

## Project Submission Requirements

### File/folder structure and naming convention

You need to create a folder named Project1_ABcXXX with A signifies the first letter of your first name, Bc signifies the first two letters of your last name and XXXXX denotes the last three digits of your student ID. The project **must be submitted as a zip file**. Other type of compression (tar.gz, tar, bz2, rar) is not acceptable. Please make sure to check whether your zip file can be unzipped and **contains all the required files for the project to work properly**. Make sure to update the path of your notebook so that it can read your dataset

**The zip file must have the following structure:**

| Name/structure | Comment |
|---|---|
| Project1_ABcXXX.zip | Submitted zip file |
| └→ Project1_ABcXXX | The project's folder |
| ├→ Project1_ABcXXX.ipynb | The Jupyter Notebook of the project (in the main folder). Make sure that your Jupyter notebook file can access the dataset file(s) |
| ├→ Module | The module's folder |
| │ └→ Module_ABcXXX.py | The module file |
| ├→ dataset | The project's report folder containing your project's dataset |
| │ └→ Your_dataset_file(s) | The dataset csv file (from Lab2) |
| ├→ report | The project's report folder containing your project document file(s) |
| │ └→ Project1_ABcXXX.docx | Your project report |

* Please see the requirements for the regression steps needed to be performed and the project report specifications in the following pages

**In order to complete this project, you must finish Lab01 and Lab02 first.** If you have not finished the lab, do that lab first since you need the dataset obtained from Lab02.

## A. Regression Modelling Requirement

You are required to create several linear regression models to predict the cars' price. In order to complete all the modeling requirements:

- You need to use all the knowledge in Python coding and/or techniques covered in the class
- You may need to do research to implement some models that may have not been covered in the class

**Jupyter Notebook requirement**
You are required to include one Jupyter file notebook in your project folder. Your Jupyter notebook should include all the markdown texts signifying the steps with correct heading, python code, comments/analysis, and/or visualizations as stated in the following instruction.

**Note:**
**You need to create the appropriate markdown headings for each section** mentioned below. Make sure to use proper heading hierarchy when you put your markdown text.

Codes should have some short comment describing the statement. Adding a markdown cell containing text before specific actions performed is appreciated. Missing these markdown results in mark reduction

**You need to create a python MODULE containing functions to be used in your project as describe in the following pages**
- You may want to start by implementing all the steps without creating module nor function
- Once your code work perfectly, you can start defining functions in your module and use it in your project
- You may need to restart the kernel and run your code again every time you modify your module file

1. **Title, Name and References**
   Include a title of your regression project. Include your name and student ID. Add information about any references you used to help complete the project
2. **Library import and data preparation**
   **Import all the required important libraries** and load the dataset you have prepared in Lab02. Have a peek of the dataset using df.head() and print out info about the dataset, e.g., the shape, the data types, etc.
   - Remove/drop any remaining null values in the dataset if there is any
   - For simplicity, rename column **log_price** into **price** and column **sqrt_mileage** to **mileage**
   - Create a new column age such that **age** is equal to **`2025 - df.year`**. After that, drop column year.
3. **Exploratory Data Analysis**
   a. Print out the summary statistics of the dataset
   b. Print out the correlation of the features.
   c. ***[Function in the module, see the note below]*** Plot the heatmap of the correlation. Note: since the dataset contains quite a number of columns, we will select the **ten columns** that have the highest correlation with the price column. You can use the following code to create a dataframe to plot the heatmap, with thresh value equal to the provided **threshold** input.

```
ind_heatmap = df.corr().price.abs().sort_values(ascending=False)[:threshold].index
df_heatmap = df[ind_heatmap]
```

   Then use the df_heatmap ONLY to create the heatmap plot. Please use threshold value anywhere between 10 and 12

d. ***[Function in the module, see the note below]*** Perform multicollinearity analysis and display the VIF data. You should print out the name of the column that has the highest VIF value, whether that column will be dropped or not from the dataframe, and df.info() after you run the VIF analysis
e. Univariate Analysis. In Lab02, you have analyzed several interesting features. For completeness, plot the distribution plot and boxplot of the **price** column. Write your observation.
f. Multivariate Analysis. Make two scatter plots to compare **price** with a few interesting features. Note, you can redo what you did in Lab 02 here.
g. Feature Observation and Hypothesis
   Provide analysis on some of the features available, and your hypothesis for the regression model. You can bring out what you have observed and your own intuition/knowledge in this part.

**Functions in the module.** Make sure you read the note in the previous page about creating the module.

a. Create a new python file **module_ABcXXX.py** within the **Modules** folder (following the naming convention)
b. Within the python module file, create a function called **`performVIF(df, drop=True)`**:
   - That will perform the VIF analysis using StandardScaler scaling method
   - Print out the column name that has the highest VIF
   - Your decision whether to drop the column following the provided input parameter **`drop`** and the highest VIF value
   - Print out the df.info() after the above operation
c. Within the python module file, create a function called **`plotHeatMap(df, threshold)`**:
   - That will select the k features with the highest correlation, with k is equal to threshold (see step 3.c above)
   - Plot the heatmap

**4. Feature Selection**

You will need to create several dataframes to store the results of the following feature selection methods to be used later in the modeling.

***a.*** Before you start, assign the **price** column of the dataframe to a new variable called **target**. Also make sure that there is no null in your dataframe. If there is still any null, make sure to use dropna to drop them.

***b. Correlation Based Selection***

Calculate the correlation between features. Since we want to predict the **price**, you want to find features that have high correlation with the **price**. Assume that you want to take any features whose correlation is greater that **0.2**. Select those features while making sure that the **price** column is not included as one of the selected. Save these new selected features as a new dataframe, for example, ***df_corr***.

***c. Select K-Best method***

Use the select K-best method to select the features. Set the value of k to be any value between **10** to **15**. Once you obtain the features, save the selected features into a new dataframe, for example, ***df_best***.

***d. Variance threshold method***

Use the variance threshold method to select the features. Set the threshold value to be any value between **0.1** to **0.15**. Once you obtain the features, save the selected features into a new dataframe, for example, ***df_var***.

5. **Linear Regression Models with Feature Selection, Feature transformation and Scaling**

In this part, you will make linear regression models by applying the following combination of feature selection, feature transformation and feature scaling methods:

- Different feature selection methods: correlation-based with threshold, selectKBest and varianceThreshold feature selection methods
- The polynomial features of degree 2 **interaction only** and without bias
- Robust scaling method

**You should use 75:25 for training and testing and random_state=42**. You should also evaluate the model by calculating the RMSE and $R^2$ metrics.

**Initial step**

In order to accumulate the result in a dataframe later, create some empty placeholder list to store your experiments' result. You should create a list to store the type of feature selection, transformation, scaling, r2 and rmse values. For example, you can create a list named as r2_scores to store all r2 scores of your models.

In order to get a full mark on this step,

a. Within the function jupyter notebook file, create a function called **ftransPoly(data)**:
   - That will perform the polynomial degree two **interaction only** of the provided data.
   - Return the transformed dataframe to the calling function
b. Avoid having a data leakage in scaling the dataframe (see https://machinelearningmastery.com/data-preparation-without-data-leakage/)
c. Use loop to perform the experiment to loop through the following combination to provide the result as shown in the following page.
   - **df_corr**, **df_best**, and **df_var**
   - Whether polynomial transform was performed or not
   - Whether the Robust scaling feature scaling method were used or not

While creating the different linear models, you may want to tract which model that gives the best result by comparing either the r2 or the rmse. You can save the information about the best model to be used for the analysis at the end. **Make sure to analyze the best model, not the last model you make!!**

***a. Linear Regression model with Correlation based feature selection***

Create a linear regression model using the correlation-based feature selection, fit, and predict the model using the test set and calculate the RMSE and $R^2$ metrics scores. Then, append the information to the lists you made at the initial step. For example, you could have something like below. Hint, you can save all the options in list or dictionary and loop through the options. For the **fSelList** list, you can use "*Correlation > threshold*" with stated threshold value (see Figure 1). Note: **minCorr** in the below is a variable that stores the minimum correlation value, e.g., 0.2., for the features to be selected.

```
# assuming that you have fSelList, fTransList and fScaleList
# that stores their corresponding model configuration
fSelList.append("Correlation > " + str(minCorr))
fTransList.append("None")
fScaleList.append("None")
```

**b. *Linear Regression model with Correlation based feature selection, no transformation, and Robust Scaler***

Create a linear regression model using the correlation-based feature selection, apply the Robust scaler, fit, and predict the model using the test set and calculate the RMSE and $R^2$ metrics scores. Then, append the information to the lists you made at the initial step. For example, you could have something like below.

```
# assuming that you have fSelList, fTransList and fScaleList
# that stores their corresponding model configuration
fSelList.append("Correlation > " + str(minCorr))
fTransList.append("None")
fScaleList.append("Robust")
```

**c. *Linear Regression model with Correlation based feature selection, Polynomial degree 2 interaction only, and no scaling***

Create a linear regression model using the correlation-based feature selection, use the polynomial degree 2 interaction only and without bias, fit, and predict the model using the test set and calculate the RMSE and $R^2$ metrics scores. Then, append the information to the lists you made at the initial step. For example, you could have something like below.

```
# assuming that you have fSelList, fTransList and fScaleList
# that stores their corresponding model configuration
fSelList.append("Correlation > " + str(minCorr))
fTransList.append("Poly 2 interaction only")
fScaleList.append("None")
```

**d. *Linear Regression model with Correlation based feature selection, Polynomial degree 2 interaction only, and Robust scaler***

Create a linear regression model using the correlation-based feature selection, use the polynomial degree 2 interaction only and without bias, apply the Robust scaler, fit, and predict the model using the test set and calculate the RMSE and $R^2$ metrics scores. Then, append the information to the lists you made at the initial step. For example, you could have something like below.

```
# assuming that you have fSelList, fTransList and fScaleList
# that stores their corresponding model configuration
fSelList.append("Correlation > " + str(minCorr))
fTransList.append("Poly 2 interaction only")
fScaleList.append("Robust")
```

**e. *Linear Regression model with SelectKBest Selection***

Use the dataframe obtained at step 4.c and repeat the task 5.a until 5.d. Make sure to always append the information to the lists you made at the initial step. For the **fSelList** list, you can use "*SelectKBest k=num*" with stated **k** number of features (see Figure 1).

**f. *Linear Regression model with VarianceThreshold Selection***

Use the dataframe obtained at step 4.d and repeat the task 5.a until 5.d. Make sure to always append the information to the lists you made at the initial step. For the **fSelList** list, you can use "*VarianceThreshold (num)*" with stated **num** as the threshold (see Figure 1).

6. **Linear Regression Model with Ridge**

You should use all the available features in this model and decides on the alpha value for the Ridge model. You need to use the same ratio for training and test and the same random state value. You can play around with the values of alpha, the number of alphas being tested, and the number iteration of the Ridge if you think you can improve your result. You should also evaluate the model by calculating the RMSE and $R^2$ metrics for each alpha value. Then you need to select the alpha value that gives you the minimum RMSE. Hint: try to sort the result and find the alpha value instead of noting down the value of alpha. After that make sure to always append the information to the lists you made at the initial step of task 6. For the fSelList list, use "Ridge, alpha =X", with X states the chosen alpha.

| | Feature Selection | Feature Transformation | Feature Scaling | R2 | RMSE |
|---|---|---|---|---|---|
| 0 | Correlation > 0.2 | None | None | | |
| 1 | Correlation > 0.2 | None | Robust | | |
| 2 | Correlation > 0.2 | Poly2 | None | | |
| 3 | Correlation > 0.2 | Poly2 | Robust | | |
| 4 | SelectKBest k=10 | None | None | | |
| 5 | SelectKBest k=10 | None | Robust | | |
| 6 | SelectKBest k=10 | Poly2 | None | | |
| 7 | SelectKBest k=10 | Poly2 | Robust | | |
| 8 | Variance Threshold (0.1) | None | None | | |
| 9 | Variance Threshold (0.1) | None | Robust | | |
| 10 | Variance Threshold (0.1) | Poly2 | None | | |
| 11 | Variance Threshold (0.1) | Poly2 | Robust | | |
| 12 | Ridge Alpha = | None | None | | |

**Figure 1: Results from all models**

7. **Plot and summary analysis**

You should combine all the information stored in the different lists into a dataframe (see Figure 1). You can either use the np.vstack() function or use list and zip functions to create the dataframe summarizing all your experiments. You should have 13 different models. The dataframe should display that information as shown in Figure 1. Note that the value of RMSE and $R^2$ metrics in the figure above are hidden. The alpha for the Ridge models shown in the dataframe above are the alpha (also shaded) that gives the smallest RMSE for each corresponding model. Based on the result dataframe above, select the best linear model and make the prediction. **Plot the scatter plot** to compare the output of your model (Y_pred) and the test dataset (Y_test). Then please make sure to **print the coefficient** for the best linear model. **Make comments on your findings**. Make sure to comment why do you think a specific linear model, or feature selection method perform better than the others.

8. **Out of Sample Prediction**
   Create a synthetic dataset containing at least two rows of data points that has the same columns set as the one that gives the best linear regression. You can look at the df.describe() and choose one of the rows, e.g., mean, 5% percentile, 75% percentile, etc. However, make sure that:
   - All values are numerical
   - Your synthetic dataset does not have conflicting information. For example, the car cannot have a value of one for both fl_electric and fl_petrol, or mk_ford and mk_audi.

   Once you created the synthetic dataset, make sure to transform or scale this dataset if your best linear regression requires you to transform or scale. Comments on the result of your prediction.

## B. Project Report Requirement

Based on your Jupyter notebook code and result, you should create a project report containing your findings. The document report should not exceed 6 pages. **DO NOT create any title page**. Just state your name and student ID at the header or at the top of the report.

The report should contain the following:

1. **Title and Introduction**
   Provide a title and short introduction about the project and dataset
2. **Dataset Analysis**
   Provide a little sneak peak of the dataset. Briefly explain the summary of data preparation performed (from Lab02) and the shape of final dataframe to be used for the modelling.
3. **EDA**
   Provide plots of some interesting features. You should generate the plot in your Jupyter notebook and embed it here. You should display the correlation among features. Make sure to make comments and observations on the visualizations. It is better to put a few visualizations with clear observation than a lot of visualization with no observation.
4. **Feature observation and hypothesis**
   Provide a brief paragraph from your simple observation of the features and also state your hypothesis for the regression model.
5. **Linear Regression Report**
   Briefly explain the different feature engineering methods you employed to create the linear regression model. For each of the feature selection methods used, provide the list of selected features you get. For the correlation based selection, make sure to state why you select those features. Also provide the result of your regression model: the features used, the alpha and other parameters used, the performance metrics of the model.
6. **Analysis**
   Embed the table and plot you obtained in the last step of the Jupyter Notebook requirement. Write a few paragraphs containing your observation on the result and some suggestion on different steps/methods you think could be taken to improve the quality of your machine learning prediction.

**C. Project Grading Criteria**

The project will be graded on a scale of 30 points.

| Criteria | | Grading |
|---|---|---|
| Jupyter | The code produces error(s) and/or messily written. The code shows the student's lack of understanding of the assigned task in the project. | -20 until -26 |
| | The code is clean with no error being produced. Comments are appropriately added. | 1 |
| | Markdown texts are provided with appropriate heading and text explaining the part of the code/project. | 3 |
| | The EDA was performed perfectly along with the plots and their analysis | 3 |
| | Feature observation and hypothesis were adequately provided | 1 |
| | Linear Regression with Feature Selection and Scaling was completed perfectly.<br>• If no module were created, no loop was used, and data leakage was not avoided in performing the experiments (8 marks)<br>• If module were created, loop(s) was used, and data leakage was avoided in performing the experiments (12 marks) | Up to 12 |
| | Linear regression with Ridge was completed perfectly | 2 |
| | Table summary, plot and analysis were adequately provided | 4 |
| Report | The report is unstructured, contains only screen shots or lacking text content, written in bullet points instead of paragaphs, using bigger than commonly used font, have bad grammars, and spelling, etc. | -1 until -4 |
| | The report was submitted following the stated requirements | 4 |