DOUGLAS COLLEGE – Fall 2022

CSIS 3290 – 001 – (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 (Lab2) that contains information of tens of thousands car auction dataset. You will need to implement several linear regression techniques to predict the car price.

Project Submission Requirements

File/folder structure and naming convention

You need to create a folder named Project1_ABcXXXXX 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 five 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_ABcXXXXX.zip	Submitted zip file
└ Project1_ABcXXXXX	The project's folder
├ Project1_ABcXXXXX.ipynb	The Jupyter Notebook of the project (in the main folder). Make sure that your Jupyter notebook file can access the dataset file(s)
├ functions	The function's folder
│ └ Functions_ABcXXXXX.ipynb	The function file
├ Dataset	The project's report folder containing your project's dataset
│ └ Your_dataset_file(s)	The dataset csv file (from Lab1)
├ Report	The project's report folder containing your project document file(s)
│ └ Project1_ABcXXXXX.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 first. If you have not finished the lab, do that lab first since you need the dataset obtained from Lab01.

A. Regression Modelling Requirement

You are required to create several linear regression models to predict the car emissions. 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
- You may need to wait until some topics were covered in the next class to complete some of the tasks below

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 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

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 Lab01. Have a peek of the dataset using df.head() and print out info bout the dataset, e.g., the shape, the data types, etc.
   a. If your saved cleaned dataset from Lab01 include the index value, make sure to drop the index column
   b. Remove/drop any remaining null values in the dataset
   c. Check out the number of unique values in the model column. Since there are just too many of them and they do not follow the format as specified in the dataset documentation, please drop the model column.

3. **Exploratory Data Analysis**
   a. Print out the summary statistics of the dataset
   b. Print out the correlation of the features. Plot the heatmap of the correlation.
   c. Notice that the correlation between features fuel_comb, fuel_city and fuel_hwy (also cylinder and engine) are very high. There may be some multicollinearity between these features (will be covered in the next lecture).
      - Calculate the Variance Influence Factor (VIF) for the numerical columns of the dataframe
      - If the VIF is very high, scale a data frame with standard scaler and re-calculate the VIF. Drop the column that has the highest VIF.
        Normal VIF value should be less than or equal to five. Hint: you may want to keep just one of the three fuel efficiency features. We will keep both the cylinder and engine for now
   d. Univariate Analysis
      - Plot the distribution plot and boxplot of the **emissions** column. Write your observation.
      - Plot the distribution of the remaining fuel efficiency column. Write your observation.
      - Plot the distribution either the engine or cylinder column. Write your observation.

e. Multivariate Analysis
    - Create the scatter plot to compare the emissions with the remaining fuel efficiency column. Write your observation.
    - Create the scatter plot to compare the emissions with the engine or cylinder column. Write your observation.

f. Feature Observation and Hypothesis
   Provide analysis on 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.

4. Data Transformation (dummy features)

We need to transform the categorical column into dummy features and join them with the original dataframe. You can create the dummy features for each categorical column one by one, without creating any function in a separate Jupyter file. You will lose some marks, but you can continue with the project. Note that the number of columns after you join the dummy features should be 41.

If you want to create a function in a separate file to get full mark:

a. Create a new jupyter notebook file within the **functions** folder (following the naming convention)
b. Within the function jupyter notebook file, create a function called **make_dummies(data)**:
    - That will select all the categorical columns from the dataframe and create the dummies.
    - Drop the categorical column from the dataframe
    - Join the numerical column of the dataframe with the dummies and return the newly created dataframe
c. Make sure to run the function jupyter notebook file in your main script and then call the make_dummies() function. At this point you should have 41 columns.

5. 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 **emissions** column of the dataframe to a new variable called target, i.e., target = df.emissions. Please keep the original dataframe and work on its copy instead.

b. Correlation Based Selection (manual or threshold)

Calculate the correlation between features. Since we want to predict the emissions, you want to find features that have high correlation with the emissions. Hint: you can sort the output of correlation by the emissions column. Remember to get the absolute value of the correlation since strong negative correlation is also a good correlation. You can either:

- Select between 6 to 10 columns using your own manual observation of the correlation value
- Use a threshold value, i.e., correlation value > 0.15, to select the features you want to use.

Make sure to drop the emission column from the selected feature. Save these new selected features as a new dataframe, for example, *df_correlation*

c. Variance Threshold Selection

Use the variance threshold method to select the features. Please use the threshold value in the range of 0.1 to 0.15. Assuming that *vt* is the *VarianceThreshold* class you use to select the features, the selected feature can be obtained by using the following statement **`df.iloc[:,vt.get_support()]`**.

Make sure to drop the emission column from the selected feature. Save the selected features into a new dataframe, for example, *df_vt*

d. Select K-Best method

Please refer to the demo code for the SelectKBest method. Use the select K-best method to select the features. Please use the number of features to be selected between eight to twelve. Once you obtain the features, save the selected features into a new dataframe, for example, *df_selKBest*

6. Linear Regression Models with Feature Selection, Feature transformation and Scaling

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

- Different selected features: correlation based selection, variance threshold, and selectKBest features
- The polynomial features of degree 2 without bias
- Standard 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. Use the jupyter function file you created in the previous step when you created the dummies.
b. Within the function jupyter notebook file, create a function called **make_poly(data)**:
    - That will perform the polynomial degree of two of the provided data
    - Return the transformed dataframe to the calling function
c. Within the function jupyter notebook file, create a function called **make_scale(data)**:
    - That will perform the standard scaling of the provided data
    - Return the scaled dataframe to the calling function
d. Use loop to perform the experiment to loop through the following combination to provide the result as shown in the following page.
    - All feature selection dataframes: df_correlation, df_vt and df_selKBest
    - Whether polynomial transform was performed or not
    - Whether feature scaling was performed or not

a. Linear Regression model with the Correlation based selected features

Create a linear regression model using the correlation based selected features, 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:

```
fsel_list.append("Correlation")
fpoly_list.append("None")
fscale_list.append("None")
r2_scores.append(r2)          # assuming r2 is the variable that stores your R^2 score
rmse_scores.append(rmse)      # assuming rmse is the variable that stores your RMSE score
```

b. Linear Regression model with Correlation based selected features and Polynomial Features transformation

Create a linear regression model using the correlation based selected features and use a polynomial degree of 2 like we did in the demo code. Then fit and predict the model using the test set and calculate the RMSE and R^2 metrics scores. The column names for the polynomial features can be obtained with **`pf.get_feature_names_out(df_correlation.columns)`**, assuming that *pf* is variable for the *PolynomialFeatures* class you use. Make sure to append the following information to the lists you made at the initial step.

```
fsel_list.append("Correlation")
fpoly_list.append("Poly2")
fscale_list.append("None")
r2_scores.append(r2)          # assuming r2 is the variable that stores your R^2 score
rmse_scores.append(rmse)      # assuming rmse is the variable that stores your RMSE score
```

c. Linear Regression model with Correlation based selected features and a feature scaling method

Create a linear regression model using the correlation based selected features and use the standard scaler. Then fit and predict the model using the test set and calculate the RMSE and R^2 metrics scores. Make sure that the features used in the linear model is the scaled features of the feature set used. Then, append the following information to the lists you made at the initial step.

```
fsel_list.append("Correlation")
fpoly_list.append("None")
fscale_list.append("Standard")
r2_scores.append(r2)          # assuming r2 is the variable that stores your R^2 score
rmse_scores.append(rmse)      # assuming rmse is the variable that stores your RMSE score
```

d. Linear Regression model with Correlation based selected features, Polynomial Features and feature Scaling

Create a linear regression model using the correlation based selected features, use a polynomial degree of 2 (without bias) and the standard scaling methods. Make sure that you find the polynomial features before scaling it. Then fit and predict the model using the test set and calculate the RMSE and R^2 metrics scores. When reconstructing the dataframe to be used by the LinearRegression, you can use the get_feature_names_out() from the polynomial features class. Then, append the following information to the lists you made at initial step.

```
fsel_list.append("Correlation")
fpoly_list.append("Poly2")
fscale_list.append("Standard")
r2_scores.append(r2)          # assuming r2 is the variable that stores your R^2 score
rmse_scores.append(rmse)      # assuming rmse is the variable that stores your RMSE score
```

e. Linear Regression model with Variance Threshold Selection

Use the dataframe obtained at step 5.c and repeat the task 6.a until 6.d. Make sure to always append the information to the lists you made at the initial step. For the fsel_list list, you can use “Variance”.

f. Linear Regression model with SelectKBest Selection

Use the dataframe obtained at step 5.d and repeat the task 6.a until 6.d. Make sure to always append the information to the lists you made at the initial step. For the fsel_list list, you can use "SelectKBest".

Note regarding task number 7 (Lasso Linear Regression) below
Depending on our progress on the lecture on Tuesday 18th October, we may have covered Lasso regression on that day. Otherwise, the Lasso regression will be covered in the lecture after the Midterm.

7. Linear Regression Model with Lasso

You should use all the available features in this model and decides on the alpha value for the Lasso 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 Lasso 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 fsel_list list, use "Lasso, alpha =X", with X states the chosen alpha.

Figure showing of the result table of your experiments

	Feature Selection	Feature Transformation	Feature Scaling	R2	RMSE
0	Correlation	None	None		
1	Correlation	None	Standard Scaler		
2	Correlation	Poly degree 2	None		
3	Correlation	Poly degree 2	Standard Scaler		
4	Variance	None	None		
5	Variance	None	Standard Scaler		
6	Variance	Poly degree 2	None		
7	Variance	Poly degree 2	Standard Scaler		
8	SelectKBest	None	None		
9	SelectKBest	None	Standard Scaler		
10	SelectKBest	Poly degree 2	None		
11	SelectKBest	Poly degree 2	Standard Scaler		
12	Lasso Alpha =	None	None		

8. Plot and summary analysis

You should combine all the information stored in the different lists into a dataframe (see the figure above). 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 below. Note that the value of RMSE and R^2 metrics in the figure above are hidden. The alpha for the Lasso models shown in the dataframe above are the alpha (also shaded) that gives the smallest RMSE for each corresponding model. You may get results that are not similar to the one shown in the shown figure.

Based on the result dataframe above, you should select the best linear model. Use that linear model to make the prediction. Make sure to save the model as a variable since you are going to use it for the next task. **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.

9. **Out of Sample Prediction**

   Create a synthetic dataset containing one to two rows of data points that has the same columns set as the one that gives the best linear regression. You can put some arbitrary numbers, or you can also 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 in integers. You may need to round them
   - Your synthetic dataset does not have conflicting information. For example, the car cannot have a value of one for both make_honda and make_hyundai. You should also check for the other columns generated using the dummy features, i.e., the fuel type, transmission, etc.

   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.

10. **Negative Value**

    Even though that your linear regression model has a high accuracy and score, it may not prevent your model to give you a negative prediction!
    Find out the negative values on your Y_pred. Look at the corresponding rows from the test set that produces those negative predictions. Relate it with the coefficients of your best model. Can you make any comments or observation about it?

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 Lab01) 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. **Simple 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.

6. **Linear Regression with Lasso Report**
   Provide the result of your regression model: the features used, the alpha and other parameters used, the performance metrics of the model.

7. **Analysis**
   Embed the table and plot you obtained in the last step of the Jupyter Notebook requirement. Write a few paragraph 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. Comments and markdown texts are inadequate.	-20 until -25
	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.	2
	The EDA was performed perfectly along with the plots and their analysis	3
	Data transformation was performed correctly • If no function were created to create the dummy features (1 mark) • If a function were created at a separate jupyper notebook to create the dummy features (2 marks)	Up to 2
	Feature observation and hypothesis were adequately provided	1
	Linear Regression with Feature Selection and Scaling was completed perfectly. • If no functions were created at a separate jupyper notebook and no loop was used to perform the experiments (8 marks) • If functions were created at a separate jupyter notebook and loop was used to perform the experiments (12 marks)	Up to 12
	Linear regression with Lasso was completed perfectly	2
	Table summary, plot and analysis were adequately provided	4
Report	The report is unstructured, contains only screen shots and lacking text content, written in the bullet points, using bigger than commonly used font, have bad grammars, and spelling, etc.	-1 until -3
	The report was submitted following the stated requirements	3