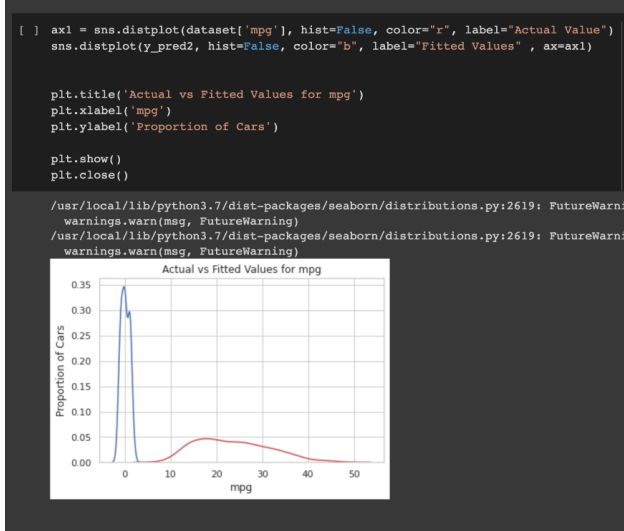


Project Development Phase Model Performance Test

Date	14 November 2022
Team ID	PNT2022TMID05997
Project Name	Project - Machine Learning based Vehicle Performance Analyser
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in the model performance testing template.

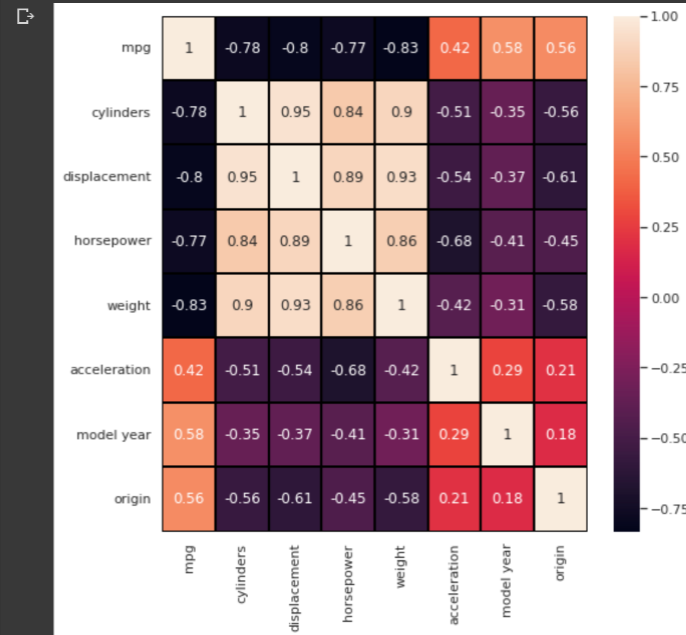
S.No.	Parameter	Screenshot / Values
1.	Dashboard design	<p>No of Visualizations / Graphs -</p>  <pre>[] ax1 = sns.distplot(dataset['mpg'], hist=False, color="r", label="Actual Value") sns.distplot(y_pred2, hist=False, color="b", label="Fitted Values" , ax=ax1) plt.title('Actual vs Fitted Values for mpg') plt.xlabel('mpg') plt.ylabel('Proportion of Cars') plt.show() plt.close()</pre> <p>/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning warnings.warn(msg, FutureWarning) /usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning warnings.warn(msg, FutureWarning)</p> <p>Actual vs Fitted Values for mpg</p> <p>The figure is a line plot titled 'Actual vs Fitted Values for mpg'. The x-axis is labeled 'mpg' and ranges from 0 to 50. The y-axis is labeled 'Proportion of Cars' and ranges from 0.00 to 0.35. There are two data series: 'Actual Value' represented by a red line and 'Fitted Values' represented by a blue line. The red line shows a sharp peak at approximately mpg=0 with a proportion of about 0.35. The blue line shows a broader distribution starting around mpg=10, peaking at approximately mpg=20 with a proportion of about 0.05, and then tapering off towards mpg=50.</p>

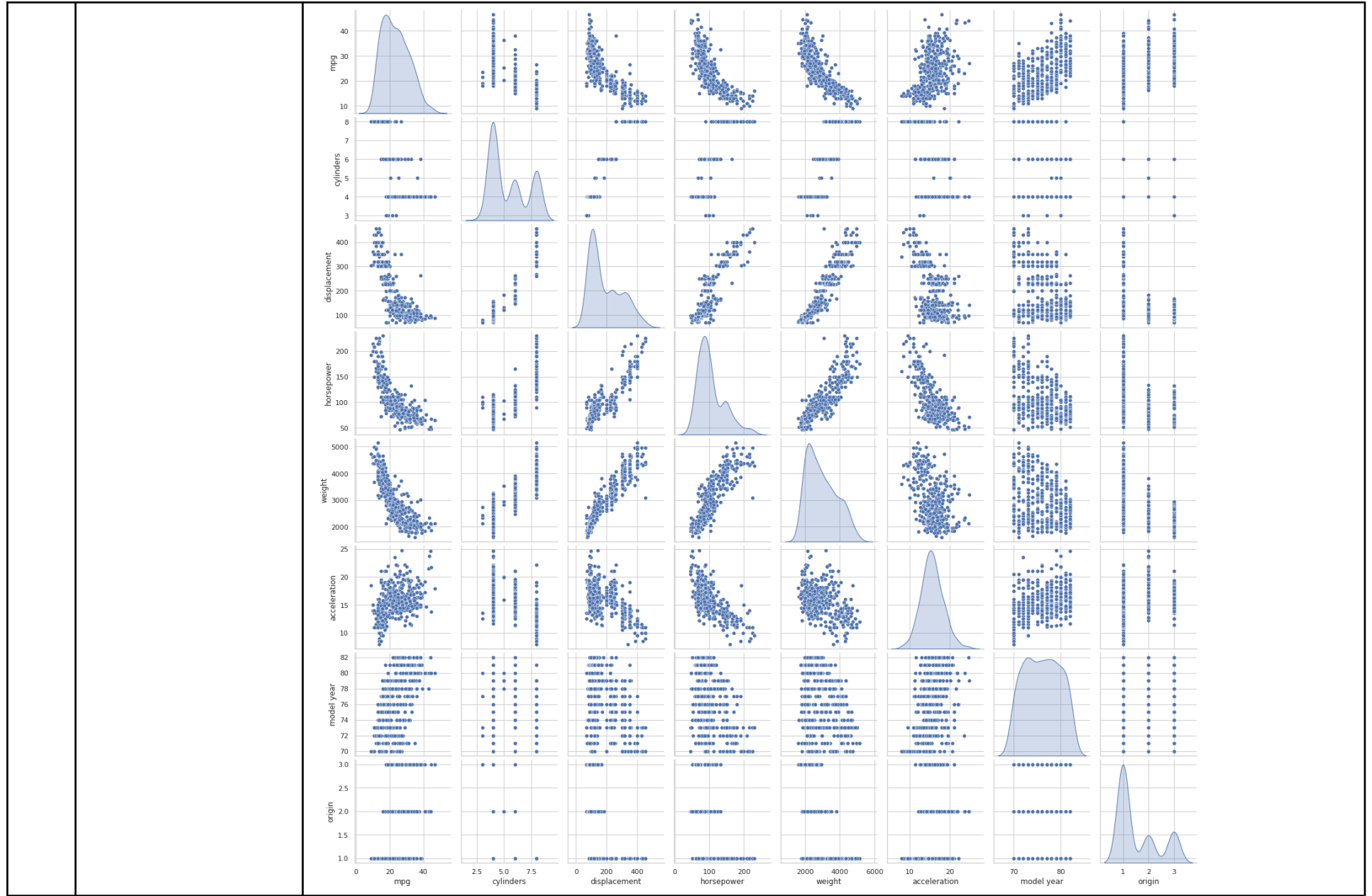
2.	Data Responsiveness	<pre> 0 mpg 398 non-null float64 1 cylinders 398 non-null int64 2 displacement 398 non-null float64 3 horsepower 398 non-null float64 4 weight 398 non-null int64 5 acceleration 398 non-null float64 6 model year 398 non-null int64 7 origin 398 non-null int64 8 car name 398 non-null object </pre>
3.	Amount Data to Rendered (DB2 Metrics)	<pre> RangeIndex: 398 entries, 0 to 397 </pre>
4.	Utilization of Data Filters	Standardscaler
5.	Effective User Story	<p>No of Scene Added - 5</p> <pre> Worst performance with mileage Low performance with mileage Medium performance with mileage High performance with mileage Very high performance with mileage </pre>
6.	Descriptive Reports	No of Visualizations / Graphs -

▼ Data Visualizations

Heatmap : which represents correlation between attributes

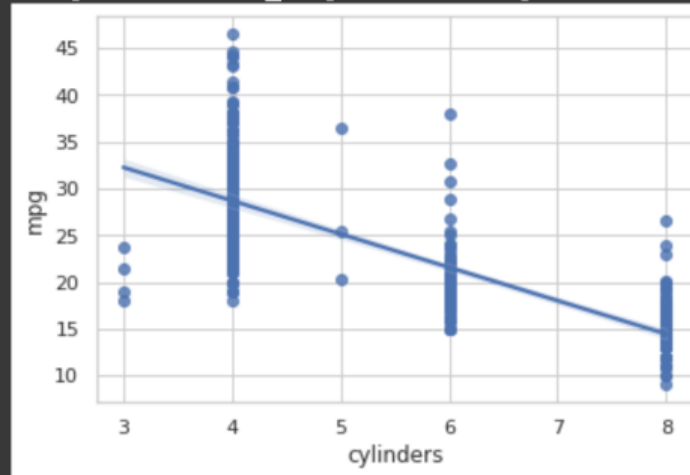
```
sns.heatmap(dataset.corr(),annot=True,linecolor='black', linewidths = 1)
fig=plt.gcf()
fig.set_size_inches(8,8)
```





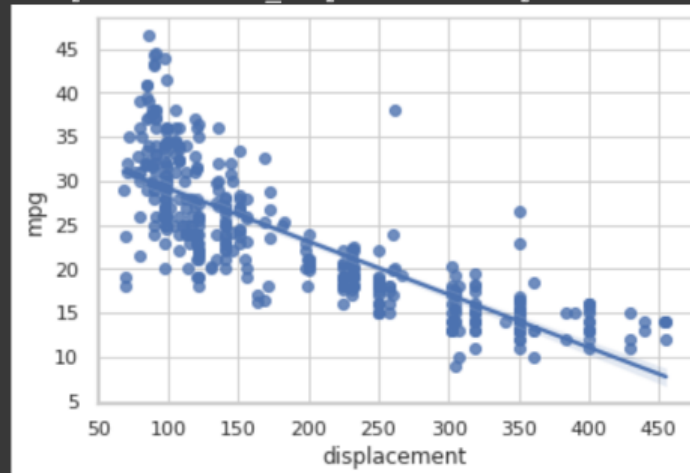
```
[ ] sns.regplot(x="cylinders", y="mpg", data=dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5df3576cd0>
```



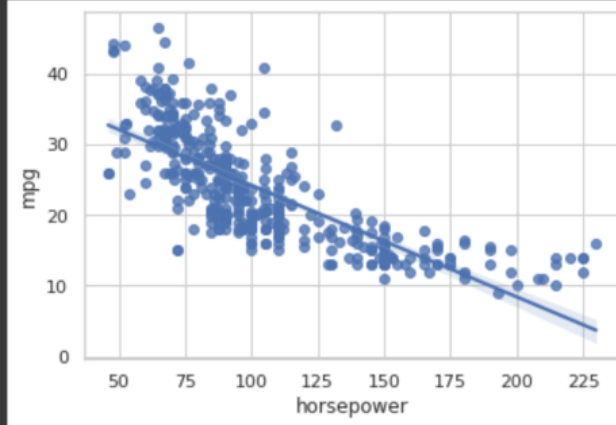
```
▶ sns.regplot(x="displacement", y="mpg", data=dataset)
```

```
☐ <matplotlib.axes._subplots.AxesSubplot at 0x7f5df3431150>
```



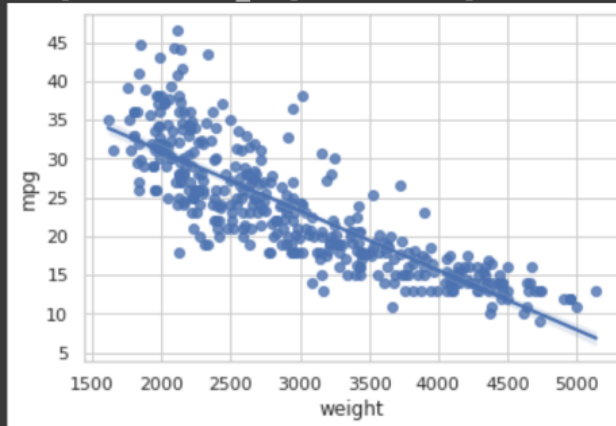
```
sns.regplot(x="horsepower", y="mpg", data=dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5df3410a90>
```



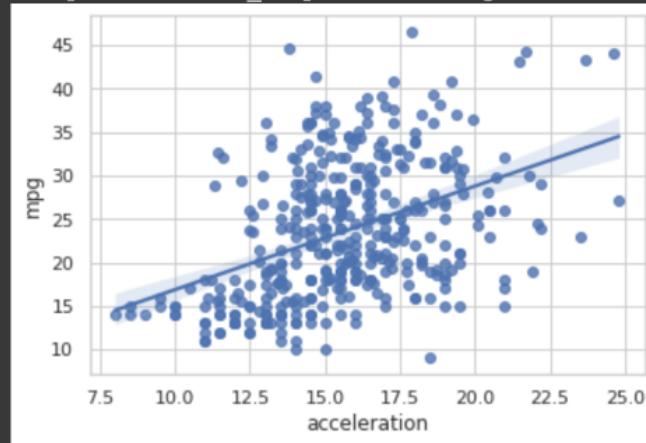
```
[ ] sns.regplot(x="weight", y="mpg", data=dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5df339ed90>
```



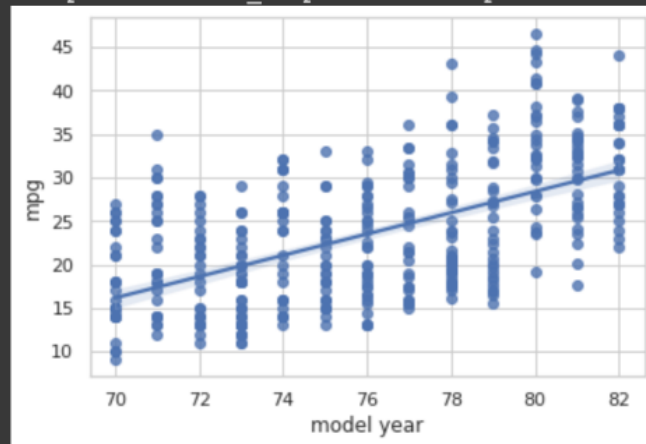
```
▶ sns.regplot(x="acceleration", y="mpg", data=dataset)
```

```
↳ <matplotlib.axes._subplots.AxesSubplot at 0x7f5df330cd90>
```



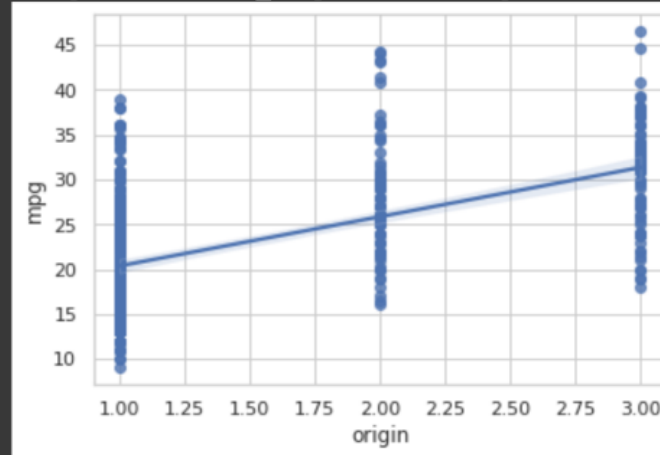
```
[ ] sns.regplot(x="model year", y="mpg", data=dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5df3295290>
```



```
▶ sns.regplot(x="origin", y="mpg", data=dataset)
```

```
↳ <matplotlib.axes._subplots.AxesSubplot at 0x7f5df31f9890>
```



```
[ ] sns.set(style="whitegrid")  
sns.boxplot(x=dataset["mpg"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5df3182b10>
```

