Data Collection and Preprocessing Phase

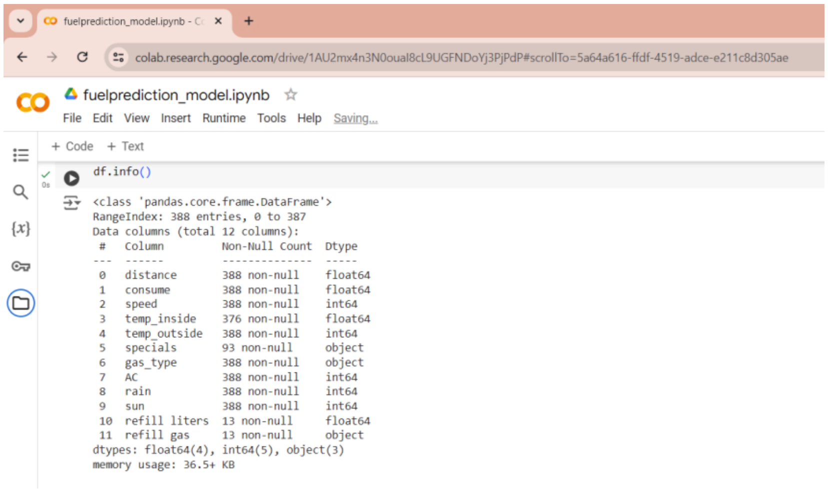
Date	9 July 2024
Team ID	739659
Project Title	Trip-Based Modelling of Fuel Consumption in Modern Fleet Vehicles Using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>

388 rows x 12 columns, dtypes: float64(4),int64(5),object(3)

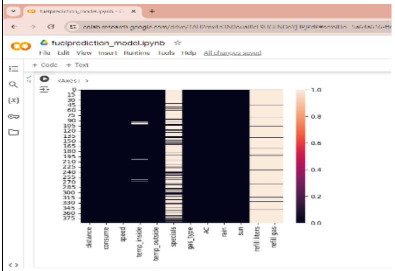

fuelprediction_model.ipynb - C
colab.research.google.com/drive/1AU2mx4n3N0oual8cL9UGFNDoYj3PjPdP#scrollTo=5a64a616-ffdf-4519-adce-e211c8d305ae
fuelprediction_model.ipynb
File Edit View Insert Runtime Tools Help Saving...
+ Code + Text
0s

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   distance       388 non-null    float64
 1   consume        388 non-null    float64
 2   speed          388 non-null    int64
 3   temp_inside    376 non-null    float64
 4   temp_outside   388 non-null    int64
 5   specials       93 non-null     object
 6   gas_type       388 non-null    object
 7   AC             388 non-null    int64
 8   rain           388 non-null    int64
 9   sun            388 non-null    int64
 10  refill liters  13 non-null     float64
 11  refill gas     13 non-null     object
dtypes: float64(4), int64(5), object(3)
memory usage: 36.5+ KB
```

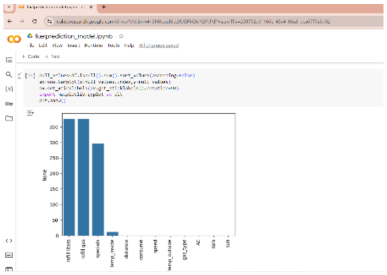
</td></tr>
<tr><td>Univariate Analysis</td><td>Exploration of individual of accuracy_score,mean_squared_error,r2_score,mean_absolute_error</td></tr>
<tr><td>Bivariate Analysis</td><td>Relationships between two variables (correlation, scatter plots) .</td></tr>
</table>

<table>
<tr>
<td></td>
<td>



fuelprediction_model.ipynb

```
<Axes: >
```

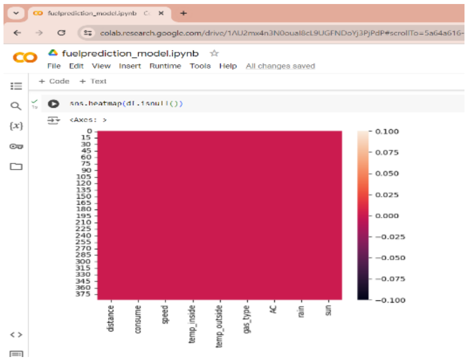


fuelprediction_model.ipynb

```
null_values=df.isnull().sum().sort_values(ascending=False)
ax=sns.barplot(x=null_values.index,y=null_values)
ax.set_xticklabels(ax.get_xticklabels(),rotation=90)
import matplotlib.pyplot as plt
plt.show()
```

</td>
</tr>
<tr>
<td>Multivariate Analysis</td>
<td></td>
</tr>
<tr>
<td>Outliers and Anomalies and null values</td>
<td>

We do not have any outliers and analomalies in our dataset except that we have null values so we have handle the null values and drop them in the way as follows



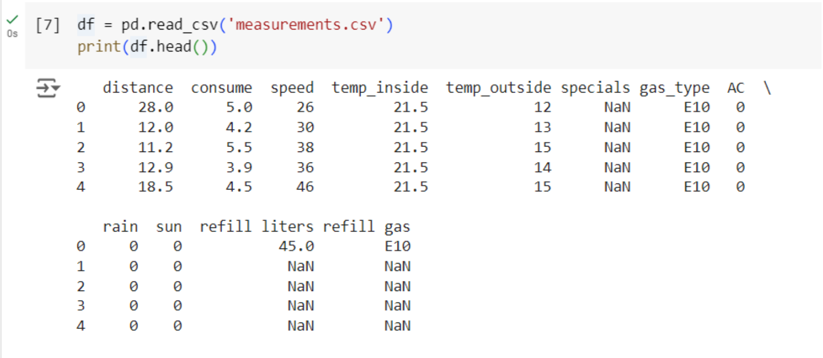
fuelprediction_model.ipynb

```
sns.heatmap(df.isnull())
<Axes: >
```

</td>
</tr>
<tr>
<td colspan="2">

Data Preprocessing Code Screenshots

</td>
</tr>
<tr>
<td>Loading Data</td>
<td>



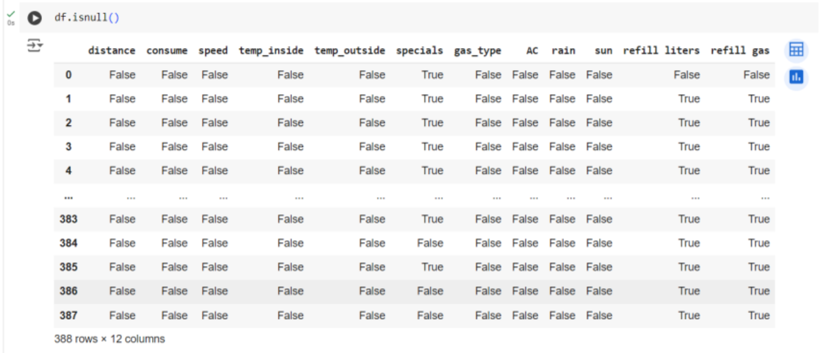
```
[7] df = pd.read_csv('measurements.csv')
    print(df.head())

     distance  consume  speed  temp_inside  temp_outside specials gas_type  AC  \
  0      28.0      5.0     26         21.5            12      NaN      E10   0
  1      12.0      4.2     30         21.5            13      NaN      E10   0
  2      11.2      5.5     38         21.5            15      NaN      E10   0
  3      12.9      3.9     36         21.5            14      NaN      E10   0
  4      18.5      4.5     46         21.5            15      NaN      E10   0

     rain  sun  refill liters refill gas
  0     0    0           45.0        E10
  1     0    0            NaN        NaN
  2     0    0            NaN        NaN
  3     0    0            NaN        NaN
  4     0    0            NaN        NaN
```

</td>
</tr>
</table>

<table>
<tr>
<td>Handling Missing Data</td>
<td>



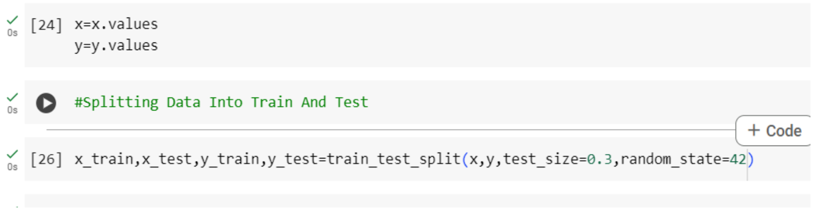
```
df.isnull()
```

	distance	consume	speed	temp_inside	temp_outside	specials	gas_type	AC	rain	sun	refill liters	refill gas
0	False	False	False	False	False	True	False	False	False	False	False	False
1	False	False	False	False	False	True	False	False	False	False	True	True
2	False	False	False	False	False	True	False	False	False	False	True	True
3	False	False	False	False	False	True	False	False	False	False	True	True
4	False	False	False	False	False	True	False	False	False	False	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...
383	False	False	False	False	False	True	False	False	False	False	True	True
384	False	False	False	False	False	False	False	False	False	False	True	True
385	False	False	False	False	False	True	False	False	False	False	True	True
386	False	False	False	False	False	False	False	False	False	False	True	True
387	False	False	False	False	False	False	False	False	False	False	True	True

388 rows × 12 columns

</td>
</tr>
<tr>
<td>Data Transformation</td>
<td>

Splitting the data :



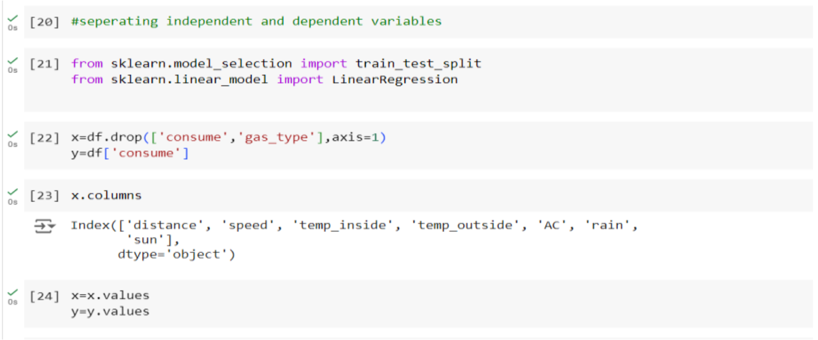
```
[24] x=x.values
     y=y.values
```

```
#Splitting Data Into Train And Test
```

```
[26] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=42)
```

</td>
</tr>
<tr>
<td>Feature Engineering</td>
<td>

modifying existing ones



```
[20] #seperating independent and dependent variables
```

```
[21] from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
```

```
[22] x=df.drop(['consume','gas_type'],axis=1)
     y=df['consume']
```

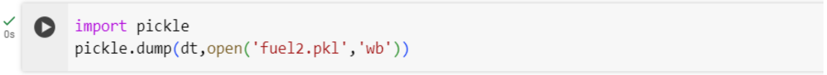
```
[23] x.columns
```

```
Index(['distance', 'speed', 'temp_inside', 'temp_outside', 'AC', 'rain',
       'sun'],
      dtype='object')
```

```
[24] x=x.values
     y=y.values
```

</td>
</tr>
<tr>
<td>Save Processed Data</td>
<td>

Code to save the cleaned and processed data for future use.



```
import pickle
pickle.dump(dt,open('fuel2.pkl','wb'))
```

</td>
</tr>
</table>