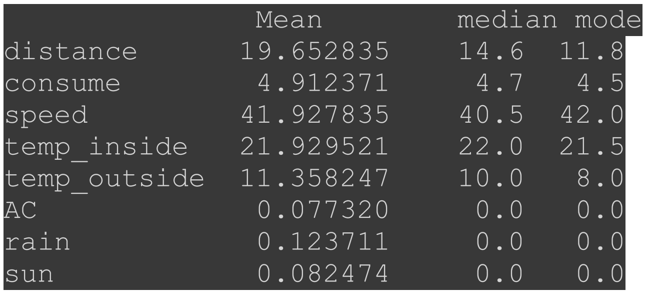
Data Collection and Preprocessing Phase

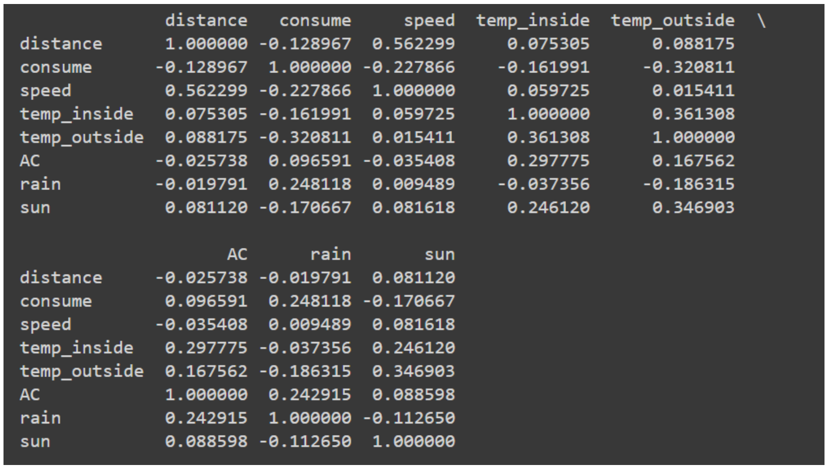
Date	15 July 2024
Team ID	SWTID1720090652
Project Title	Predictive Modelling for Fleet Fuel Management using Machine Learning
Maximum Marks	6 Marks

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.

Section	**Description**
Data Overview	Basic statistics, dimensions, and structure of the data. The dataset contains 488 records with 12 column names (12 prospective features before cleaning). The columns are distance, consume, speed, temp_inside, temp_outside, specials, gas_type, AC, rain, sun, refill liters (contains majority null values), refill gas (contains majority null values)
Univariate Analysis	The mean, median and mode of the corresponding columns are as follows:  Mean median mode distance 19.652835 14.6 11.8 consume 4.912371 4.7 4.5 speed 41.927835 40.5 42.0 temp_inside 21.929521 22.0 21.5 temp_outside 11.358247 10.0 8.0 AC 0.077320 0.0 0.0 rain 0.123711 0.0 0.0 sun 0.082474 0.0 0.0

Bivariate Analysis



```
              distance   consume     speed  temp_inside  temp_outside  \
distance      1.000000 -0.128967  0.562299     0.075305      0.088175   
consume      -0.128967  1.000000 -0.227866    -0.161991     -0.320811   
speed         0.562299 -0.227866  1.000000     0.059725      0.015411   
temp_inside   0.075305 -0.161991  0.059725     1.000000      0.361308   
temp_outside  0.088175 -0.320811  0.015411     0.361308      1.000000   
AC           -0.025738  0.096591 -0.035408     0.297775      0.167562   
rain         -0.019791  0.248118  0.009489    -0.037356     -0.186315   
sun           0.081120 -0.170667  0.081618     0.246120      0.346903   

                    AC      rain       sun  
distance     -0.025738 -0.019791  0.081120  
consume       0.096591  0.248118 -0.170667  
speed        -0.035408  0.009489  0.081618  
temp_inside   0.297775 -0.037356  0.246120  
temp_outside  0.167562 -0.186315  0.346903  
AC            1.000000  0.242915  0.088598  
rain          0.242915  1.000000 -0.112650  
sun           0.088598 -0.112650  1.000000  
```

Multivariate Analysis

On performing feature selection with SelectKBest and f_regression, we get the following result:

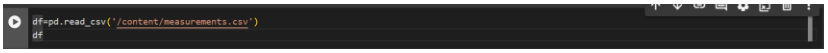
```
Selected features: Index(['speed',
'temp_inside', 'temp_outside', 'rain', 'sun'],
dtype='object')
```

Outliers and Anomalies

Identification is done by using the IQR method. We define the first and third quartile's boundaries and check for the values that lie outside and inside the boundaries.

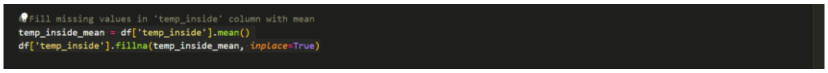
Data Preprocessing Code Screenshots

Loading Data



```
df=pd.read_csv('/content/measurements.csv')
df
```

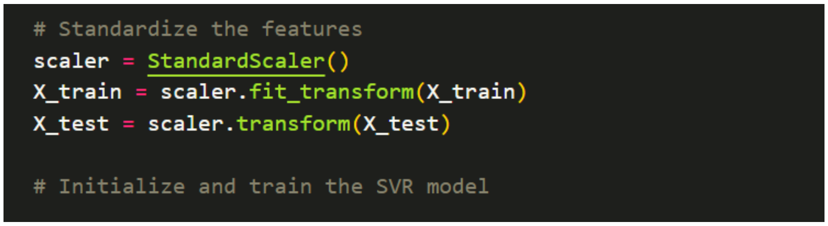
Handling Missing Data



```
#Fill missing values in 'temp_inside' column with mean
temp_inside_mean = df['temp_inside'].mean()
df['temp_inside'].fillna(temp_inside_mean, inplace=True)
```

Data Transformation

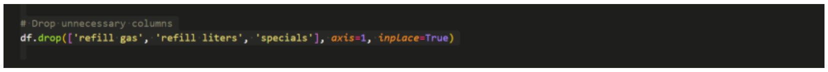
Code for transforming variables (scaling, normalization).



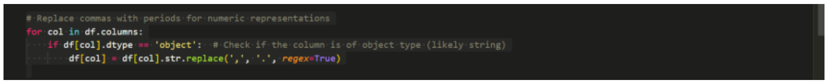
```
# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize and train the SVR model
```

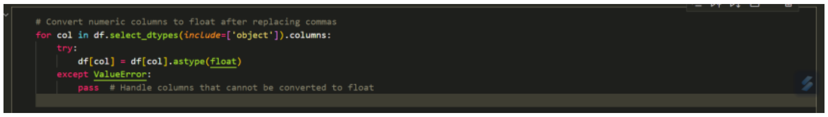
<table>
<tr>
<td>Feature Engineering</td>
<td>



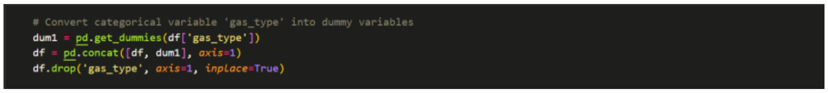
```python
# Drop unnecessary columns
df.drop(['refill gas', 'refill liters', 'specials'], axis=1, inplace=True)
```



```python
# Replace commas with periods for numeric representations
for col in df.columns:
    if df[col].dtype == 'object':  # Check if the column is of object type (likely string)
        df[col] = df[col].str.replace(',', '.', regex=True)
```

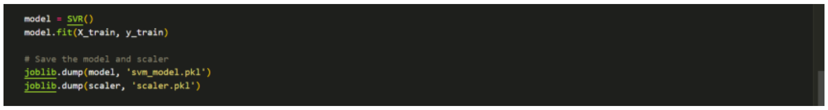


```python
# Convert numeric columns to float after replacing commas
for col in df.select_dtypes(include=['object']).columns:
    try:
        df[col] = df[col].astype(float)
    except ValueError:
        pass  # Handle columns that cannot be converted to float
```



```python
# Convert categorical variable 'gas_type' into dummy variables
dum1 = pd.get_dummies(df['gas_type'])
df = pd.concat([df, dum1], axis=1)
df.drop('gas_type', axis=1, inplace=True)
```

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



```python
model = SVR()
model.fit(X_train, y_train)

# Save the model and scaler
joblib.dump(model, 'svm_model.pkl')
joblib.dump(scaler, 'scaler.pkl')
```

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