**Introduction**

In this assignment we have a data of different cars that will allow us to make predictions based on the different factors such as the Horsepower, weight , model year, etc. With the help of these we can check which factors are affecting the milage of the car. Along with the we also we would also be able to make recommendations to build a future car. The data has 398 rows and 8 columns. This gives us limited information to make recommendations.

With the help of basic visuals, we can deduce the factors that are affecting the milage of the car and predict the factors that might affect the development of the future car.

**Data cleaning**

The data has 398 rows and 8 columns. Based on the checking, there is no missing value in the dataset. Instead of missing values we have ‘?’ that will cause inconsistencies in the data during the modeling phase.

Text

Description automatically generatedFigure Inconsistencies

Since the number of inconsistencies are really few, we can ignore these values and drop the rows that are affected by these.

As we can see there are more than 6 entities where we have ‘?’ as an entry. Since we have more than 400 records, we can drop the rows that have these entries.

**EDA**

In this section we talk about the exploratory data analysis that allows us to get a better understanding of the story the data is telling us and how different factors affect the milage of the car. in the data set.**Graphical user interface, text

Description automatically generated**

Figure Histogram.

To get a better understanding we plot a histogram of all the columns. With the help of this histogram, we can get an understanding of the variables that help in the modeling and how we would be able to tackle these. This also allows us to get an understanding of how the correlation matrix would form itself to get a better relation. Then we create a correlation matrix to get the understanding of the different variables.

Table

Description automatically generated

Figure Correlation Matrix

The understanding of the correlation matrix is made better using with the help of a heatmap which is below. The heat allows us to see what are the variables that are strongly correlated to the target variable and how these would shape. The visual analysis of the data allows us to get a understanding of how the model will change with respect to the high collinearity. With the heatmap we observe that cylinders, displacement, horsepower, weight are strongly correlated to MPG whereas Model Year and US made are moderately correlated. Chart, histogram

Description automatically generated

Figure Correlation Heatmap

**Analysis**

**Modeling:**

For the modeling process we are using the Generalized Linear Model. The linear model help us understand how scalable are the values that are included in the model and what are the different variables that are allocated for the development of the model. Text

Description automatically generated

Figure Initial VIF Score

Here we are seeing that there is high multi collinearity between the variables that are included in the model hence to deal with this issue we would be dropping those values with a high variance inflation factor or VIF. This high VIF value causes the model to have a high r2 value and get all the prediction to be incorrect. With that information we would be dropping ‘Cylinder’ ,‘Weight’ ,‘Model Year’ and ‘Displacement’ .We conduct another VIF test on the remaining values. Graphical user interface, text

Description automatically generated

Figure After Dropping

After this we see that the VIF Value reduces, and we create a model using Acceleration and US made. We create a model with the variable Acceleration and US made out which Acceleration is the feature that would be considered. Here in the model, we observe that US made and Acceleration have a high significance with MPG and we would consider acceleration to be target metric that would influence the MPG as it is more scalable compared to the.Table

Description automatically generated

Figure Model

**Conclusion**

To conclude I would say that Acceleration has a positive influence on the mileage of the cars. The fact that these are made in USA and have as a negative influence on the milage of the as it is known fact that counties such as Germany, Italy have excelled in advancement of making fuel efficient cars. The acceleration metric is which as an almost 1 to 1 relationship between which means that higher the acceleration higher would be the MPG, but it would be as scalable as US made cars.

* Recommendation to the company while developing the car would be: Understanding what the target audience would be as switching to cars that are electric in nature would completely nullify the miles per gallon concept. Here the metric of acceleration would significant as driving at a speed of economy would improve drastically what we are aiming for.
* Apart from that concentrating on production companies that aren’t US based(i.e., BMW, Volkswagen ,etc.) , are beneficial as they would improve provide better of equipment for creation of the car with better current technologies,

**Reference:**

* Seaborn.displot#. seaborn.displot - seaborn 0.12.1 documentation. (n.d.). Retrieved November 13, 2022, from <https://seaborn.pydata.org/generated/seaborn.displot.html>
* Zach. (2021, October 21). A guide to multicollinearity &amp; VIF in regression. Statology. Retrieved November 13, 2022, from <https://www.statology.org/multicollinearity-regression/>

**Appendix 1**

**Code:**

#!/usr/bin/env python

# coding: utf-8

# In[1]:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from sklearn.model\_selection import train\_test\_split

import numpy as np

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

from statsmodels.api import GLM

import statsmodels.api as sm

import warnings

warnings.filterwarnings('ignore')

# In[2]:

data = pd.read\_csv("C:/Users/kgrat/OneDrive/Documents/ALY 6020/car.csv")

# ln[3]:

data.info()

# In[4]:

df = pd.DataFrame(data)

# In[5]:

df.info()

# In[6]:

df.head()

# In[7]:

df.isnull().count()

# In[8]:

df.isna().count()

# In[9]:

df[df == '?'].count()

# In[10]:

df.Horsepower = df.Horsepower.replace('?','NaN').astype(float)

df.Horsepower.fillna(df.Horsepower.mean(),inplace=True)

df.horsepower = df.Horsepower.astype(int)

# In[11]:

df[df == '?'].count()

# In[12]:

df.describe()

# In[13]:

df.columns

# In[14]:

plt.figure(figsize=(20,20))

df.plot(kind = 'hist',subplots = 'True',layout = (4,4),figsize= (17,17))

# In[15]:

c = df.corr()

# In[16]:

c

# In[17]:

mask = np.triu(np.ones\_like(df.corr(), dtype=bool))

# In[18]:

plt.figure(figsize=((7,7)))

sns.heatmap(c,annot=True, mask=mask, vmin=-1, vmax=1)

# In[19]:

temp = pd.Series([variance\_inflation\_factor(df.values, i) for i in range (df.shape[1])], index = df.columns)

temp

# In[20]:

df\_new = df.drop(['Cylinders','Displacement','Weight','MPG','Model Year'],axis=1) #Dropped because of high nulticolinearity

# In[21]:

temp1 = pd.Series([variance\_inflation\_factor(df\_new.values,i) for i in range(df\_new.shape[1])],index=df\_new.columns)

temp1 # here we see that US made and Model Year are highly collienar

# In[22]:

X = df[['US Made','Acceleration']]

y = df['MPG']

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=.30, random\_state=123)

# In[23]:

mod = LinearRegression()

mod.fit(X\_train,y\_train)

pred = mod.predict(X\_test)

# In[24]:

mae = mean\_absolute\_error(y\_test, pred)

mse = mean\_squared\_error(y\_test, pred)

rmse = np.sqrt(mse)

# In[25]:

print(f'Mean absolute error: {mae:.2f}')

print(f'Mean squared error: {mse:.2f}')

print(f'Root mean squared error: {rmse:.2f}')

# In[45]:

x\_cn = sm.add\_constant(X\_train)

model = sm.OLS(y\_train,x\_cn).fit()

print('Parmeters :',model.params)

# In[44]:

print('Model:',model.summary())

# In[53]:

sns.displot(model.resid,kde = True, multiple="stack");

# In[51]:

fig = plt.figure(figsize=(12,8))

fig = sm.graphics.plot\_regress\_exog(model, 'Acceleration', fig)

# In[52]:

fig = plt.figure(figsize=(12,8))

fig = sm.graphics.plot\_regress\_exog(model, 'US Made', fig)