A Forecast of The Automotive Industry and its Improvements in Environmental Sustainability and Optimization of Vehicle Design

Dylan Phan

ABSTRACT

This research is to provide a comprehensive forecast of the automotive industry, with a primary focus on advancements in environmental sustainability and vehicle design optimization. As concerns about climate change and fossil fuel depletion continue to escalate, there is a pressing need to assess the potential benefits of alternative transportation options. Hybrid vehicles and those with reduced engine displacement and cylinders have garnered considerable attention due to their ability to mitigate greenhouse gas emissions and reduce reliance on fossil fuels. In this study, a diverse range of data science techniques, including prediction models, linear regression, and time series analysis, are employed to forecast the industry's trajectory. The investigation leverages datasets sourced from the Environmental Protection Agency (EPA), encompassing information on cars in the United States from 2013 to 2022, with a specific focus on CO2 emissions as the target output to be reduced. By analyzing these datasets, the research aims to evaluate the strength of smaller engine sizes, reduced engine displacement, air aspiration methods, and fewer cylinders on various critical parameters, including emissions, miles per gallon (mpg), and overall efficiency. Utilizing data science techniques, prediction models, and time series analysis, the research seeks to uncover trends, patterns, and insights into the impact of sustainable automotive technologies on emissions reduction and fuel efficiency. Through a thorough examination of the data, the study aims to provide robust evidence supporting the benefits of hybrid vehicles and vehicles with optimized engine designs in promoting environmental sustainability. A deeper understanding of the relationship between engine design parameters and environmental impact will enable the automotive industry to steer toward more eco-friendly practices. Ultimately, the results will contribute to the formulation of informed decisions and policies that promote greener transportation options, fostering a positive impact on climate change mitigation and reduced reliance on fossil fuels.

I.INTRODUCTION

Manufacturers within the automotive industry are under increasing pressure to enhance the environmental sustainability of their products while optimizing vehicle design for efficiency and performance. This research provides a valuable model that can provide information on the automotive industry's future trajectory, enabling manufacturers to anticipate trends and adapt their strategies accordingly. By accurately forecasting advancements, manufacturers can plan research and development efforts, allocate resources efficiently, and invest in technologies that align with the projected improvements. This proactive approach can foster innovation, drive the adoption of greener technologies, and contribute to building a competitive advantage in a rapidly evolving market. Government agencies and policymakers can benefit from the research findings to shape regulations and policies that encourage and accelerate sustainable practices within the automotive industry. The ability to forecast improvements can guide the formulation of realistic and effective emission standards, incentives for adopting cleaner technologies, and infrastructure development for electric vehicles. This research can aid in setting achievable targets and establishing a roadmap for a more sustainable transportation ecosystem.

Consumers play a pivotal role in driving demand for eco-friendly vehicles. With access to a model that can forecast CO2 emissions of vehicles based on a few variables, it will be able to improve environmental sustainability and vehicle design. The automotive industry's environmental footprint is a critical concern for global sustainability efforts. Forecasting improvements in environmental sustainability can have a positive impact on reducing greenhouse gas emissions, air pollutants, and overall resource consumption. This research contributes to society's broader goals of mitigating climate change and promoting a cleaner environment. Academic researchers and industry professionals can build upon the model in this research to further explore specific areas such as energy-efficient vehicle technologies, alternative fuels, and advanced materials. The accurate forecasts provided by this research serve as a foundation for more focused and targeted studies, enabling the development of innovative solutions that align with the anticipated industry improvements. The automotive industry has a substantial economic impact, contributing to employment, trade, and economic growth. Accurate forecasting of improvements allows stakeholders to make well-informed decisions, potentially attracting investments in areas that promise high growth and sustainability. Moreover, the adoption of cleaner technologies and optimized designs can lead to cost savings for both manufacturers and consumers, fostering economic resilience and stability. In summary, this research offers actionable insights into the future trajectory of the automotive industry, benefiting manufacturers, policymakers, and society at large. By facilitating informed decision-making and strategic planning, the research contributes to a more sustainable and efficient automotive sector, aligning with broader global sustainability goals and shaping the industry's evolution towards a greener future.

II.METHODS/METHODOLOGY

Data from the US Department of Energy covering the years 2013 to 2022 was obtained. The dataset includes information on CO2 emissions, vehicle specifications, annual costs, miles per gallon, etc.

```
In [4]: # Combining data files from 2013-2022 to create one large datafile in csv format
        df_2013 = pd.read_csv("2013 FEGuide-for DOE-OK to release-no-sales-9-30-2014_updated_Mercedes_public.csv")
        df_2014 = pd.read_csv("2014-FEGuide-for-DOE-OK-to-release-no-sales-5-8-2019_updated_Mercedes_public.csv")
        df_2015 = pd.read_csv("2015 FEGuide for DOE-OK to release-no-sales-5-8-2019_updated_Mercedes_public.csv")
        df_2016 = pd.read_csv("2016 FEGuide for DOE-OK to release-no-sales-5-8-2019_Mercedes_public.csv")
        df_2017 = pd.read_csv("2017 FE Guide for DOE-release dates before 3-25-2019-no sales-9-19-2019McLarenforpublic.csv")
        df_2018 = pd.read_csv("2018 FE Guide for DOE3 -all rel dates-no-sales-3-25-2019McLarenforpublic.csv")
        df_2019 = pd.read_csv("2019-FE-Guide-for-DOE-release-dates-before-12-19-2019-no-sales-12_17_2019Koenigseggpublic.csv")
        df_2020 = pd.read_csv("2020-FE-Guide-adds-and-corrections-for-DOE-OK-for-release-no-sales-4-7-2021Koenigseggpublic.csv")
        df_2021 = pd.read_csv("2021-FE-Guide-release-dates-before-11-23-2021-no-sales-11-22-2021-for-DOE_Karmapublic.csv")
        df_2022 = pd.read_csv("2022-FE-Guide-for-DOE-release-dates-before-1-12-2023-no-sales-1-11-2023public.csv")
        df_total = pd.concat([df_2012,df_2013,df_2014,df_2015,df_2016,df_2017,df_2018,df_2019,df_2020,df_2021,df_2022])
        df_total.to_csv("USA2012-2022CARS.csv",index = False)
```

Image 1: A list of all datasets that were combined for the final dataset

Categorical variables within the dataset are converted into numerical values using predefined mappings. For instance, the categorical variable 'Air Aspiration Method Desc' is mapped to numerical values based on the degree of aspiration, and 'Regen Braking Type Desc' is encoded to represent the type of regenerative brake system used. These transformations enhance the dataset's suitability for analysis and modeling. A correlation matrix was performed to understand the relationships among variables. Using correlation, features were analyzed to determine the relationship between the feature and other features in the data file. An assumption that was made before taking the correlation data is that with turbochargers, engines will be more efficient, with regenerative braking and batteries, cars will produce less CO2 emissions, and have lower fuel costs. The 'df1' DataFrame is created by extracting the selected

features that were highly correlated such as model year, division, carline, engine displacement, cylinder count, gear count, air aspiration method, battery specifications, regenerative braking type, fuel costs, and CO2 emissions. Additionally, the features are renamed for ease of reference and analysis.

In [61]:
```
df = pd.read_csv('USA2013-2022CARS - USA2012-2022CARS.csv.csv')
```

In [62]:
```
# Filling in all null values with 0
df.fillna(0, inplace=True)
```

In [63]:
```
# Hot-encoding to represent categorical variables as numerical values
value_mapping = {
    "Turbocharged": 1,
    "Other" : 1,
    "Naturally Aspirated": 0,
    "Supercharged": 2,
    "Turbocharged+Supercharged": 3
}
value_mapping2 = {"Electrical Regen Brake":1,
                  "Other" : 1,
                  "Not applicable": 0,
                   "Hydraulic Regen Brake": 1
}
value_mapping3 = {"[=0]5":5.5,
                   "[=0]6":6.5,
                   "[=0]4":4.5
}
df["Batt Energy Capacity (Amp-hrs)"] = df["Batt Energy Capacity (Amp-hrs)"].replace(value_mapping3)
df["Air Aspiration Method Desc"] = df["Air Aspiration Method Desc"].replace(value_mapping)
df["Regen Braking Type Desc"] = df["Regen Braking Type Desc"].replace(value_mapping2)
df.to_csv('USA2013-2022CARS_-_USA2012-2022CARS.csv.csv', index=False)
df.to_csv('USA2013-2022CARS_-_USA2012-2022CARS.csv.csv', index=False)
```

Image 2: Data Cleaning and Processing

In [6]:
```
# Choosing features for new data file
df1 = df[['Model Year','Division','Carline', 'Eng Displ', '# Cyl','# Gears','Air Aspiration Method Desc','# Batteries','Total Vol
```

In [7]:
```
# Renaming for simplicity
df1 = df1.rename(columns={"Model Year": "Year",
                          "Division": "Brand",
                          "Carline": "Model",
                          "Eng Displ": "Displ",
                          "# Cyl": "Cyl",
                          "# Gears": "Gears",
                          "Air Aspiration Method Desc": "Air_Aspiration",
                          "# Batteries": "Batteries",
                          "Total Voltage for Battery Pack(s)": "Total_Voltage",
                          "Batt Energy Capacity (Amp-hrs)": "Battery_Capacity",
                          "Regen Braking Type Desc": "Regen_Brake",
                          "$ You Spend over 5 years (increased amount spent in fuel costs over 5 years - on label)": "Amt_Spent",
                          "Comb CO2 Rounded Adjusted (as shown on FE Label)": "Comb_CO2",
                          "Annual Fuel1 Cost - Conventional Fuel": "Fuel_Cost"})
```

In [10]:
```
# Viewing correlation for features using color for easier visibility
corr = df1.corr(numeric_only = True)
corr.style.background_gradient(cmap='coolwarm')
```

Out[10]:

	Year	Displ	Cyl	Gears	Air_Aspiration	Batteries	Total_Voltage	Regen_Brake	Comb_CO2	Fuel_Cost
Year	1.000000	-0.048996	-0.033926	0.328478	0.165280	0.093940	0.028938	0.081830	-0.028348	-0.349075
Displ	-0.048996	1.000000	0.923490	0.219975	-0.174348	-0.058737	-0.064493	-0.062167	0.847919	0.748792
Cyl	-0.033926	0.923490	1.000000	0.245919	-0.046457	-0.046510	-0.069371	-0.054119	0.830018	0.748984
Gears	0.328478	0.219975	0.245919	1.000000	0.290827	-0.082303	-0.152894	-0.105098	0.266298	0.150904
Air_Aspiration	0.165280	-0.174348	-0.046457	0.290827	1.000000	-0.035371	-0.105951	-0.075746	0.017575	0.038526
Batteries	0.093940	-0.058737	-0.046510	-0.082303	-0.035371	1.000000	0.805528	0.849387	-0.206637	-0.169747
Total_Voltage	0.028938	-0.064493	-0.069371	-0.152894	-0.105951	0.805528	1.000000	0.859560	-0.263489	-0.209836
Regen_Brake	0.081830	-0.062167	-0.054119	-0.105098	-0.075746	0.849387	0.859560	1.000000	-0.246561	-0.203869
Comb_CO2	-0.028348	0.847919	0.830018	0.266298	0.017575	-0.206637	-0.263489	-0.246561	1.000000	0.867483
Fuel_Cost	-0.349075	0.748792	0.748984	0.150904	0.038526	-0.169747	-0.209836	-0.203869	0.867483	1.000000

Image 3: Creating DF1 Dataframe; Correlation Matrix

The dataset is split into training and testing sets, with 60% of the data allocated for training and 40% for testing. The model is trained on the training data (X_train, Y_train), and predictions are made using the test data (X_test). The accuracy of the model is evaluated using the R2 score, which quantifies the proportion of variance in the target variable explained by the model. By implementing the LinearRegression class from the sklearn library, we established predictive relationships between CO2 emissions and attributes like engine displacement, cylinder count, gear count, etc. Accuracy was gauged using the R2 score. A score of around 86.6% was achieved using the linear regression model.

In [14]:
```
# X = features, Y = output
X = df1[['Displ','Cyl','Gears','Air_Aspiration','Batteries','Total_Voltage','Battery_Capacity','Regen_Brake','Fuel_Cost']]
Y = df1[['Comb_CO2']]
```

In [15]:
```
# Splititng dataset into train and test
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.4,random_state=100)
```

In [16]:
```
# Fitting the data and predicting
from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
linreg.fit(X_train, Y_train)
Y_pred = linreg.predict(X_test)
```

In [17]:
```
print (Y_pred) # Predicted values
print(Y_test)
```

```
[[452.76172111]
 [449.27500409]
 [345.40903107]
 ...
 [475.54549322]
 [525.72504872]
 [438.45953112]]
      Comb_CO2
3048       413
5715       589
51         310
2918       324
4835       501
...        ...
11290      313
6272       401
11012      505
3180       478
8125       521

[4865 rows x 1 columns]
```

In [18]:
```
# Calculating r2 score for accuracy
r2_score = linreg.score(X_test,Y_test)
print("Accuracy:",r2_score*100,'%')
```

```
Accuracy: 86.59459713146157 %
```

Image 4: Linear Regression

Following the linear regression model and results, Time series analysis is conducted to uncover temporal patterns and trends in CO2 emissions data. The model year is converted into a datetime format to facilitate compatibility with time series models. The time series data is set with the model year as the index and saved as a new CSV file. The analysis involves the fitting of Seasonal ARIMA (AutoRegressive Integrated Moving Average) models to capture seasonality, trend, and noise in the data.

```
In [21]: # Converting the year to be in data format to work with time series model
         import datetime
         df1['Year'] = df1['Year'].apply(lambda x: pd.to_datetime(str(x), format='%Y-%m-%d'))

In [22]: # Setting the model year as the index
         df2 = df1[['Year','Comb_CO2']].set_index(['Year'])
         # Saving the dataset as a new csv file to be able to parse out year
         df2.to_csv('test.csv')

In [23]: # Created function to parse out the year
         def parser(x):
             return pd.datetime.strptime(x, '%Y-%m-%d')
         y = pd.read_csv('test.csv', header=0, index_col=0, parse_dates=True, squeeze=True, date_parser=parser)

In [24]: # Setting each time period to begin at year end
         y.index = y.index.to_period('Y')

In [11]: # Accounting for seasonality, trend, and noise in the data
         p = d = q = range(0, 2)
         pdq = list(itertools.product(p, d, q))
         seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
         print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
         print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
         print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
         print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

5: Setting up Time Series Model

The code employs the autocorrelation_plot function from pandas.plotting to visualize the autocorrelation of the time series data. This plot aids in identifying potential autocorrelation patterns at different lags, helping determine appropriate parameters (p and q) for the ARIMA model.

```
In [53]: # Plotting autocorrelation of the data
         autocorrelation_plot(y)
         pyplot.show()
```

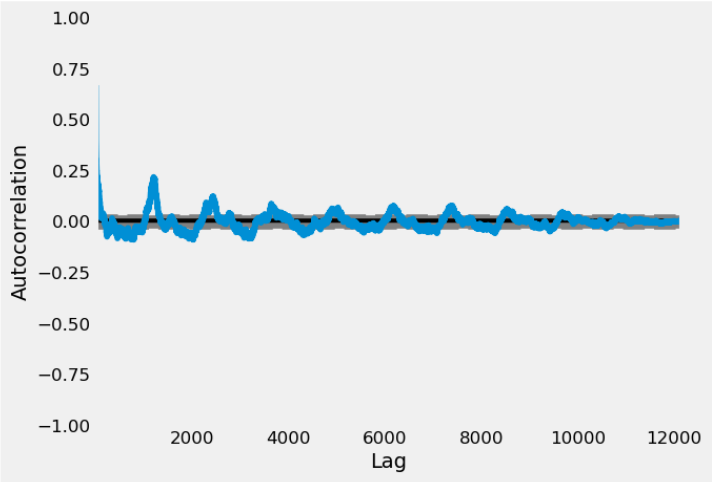


Image 6: Autocorrelation Plot 2013-2022

2022 autocorrelation plots were made to be able to view the graphs more efficiently.

```
In [70]: # Plotting autocorrelation of exact year
         autocorrelation_plot(y['2022'][0:20])
Out[70]: <Axes: xlabel='Lag', ylabel='Autocorrelation'>
```

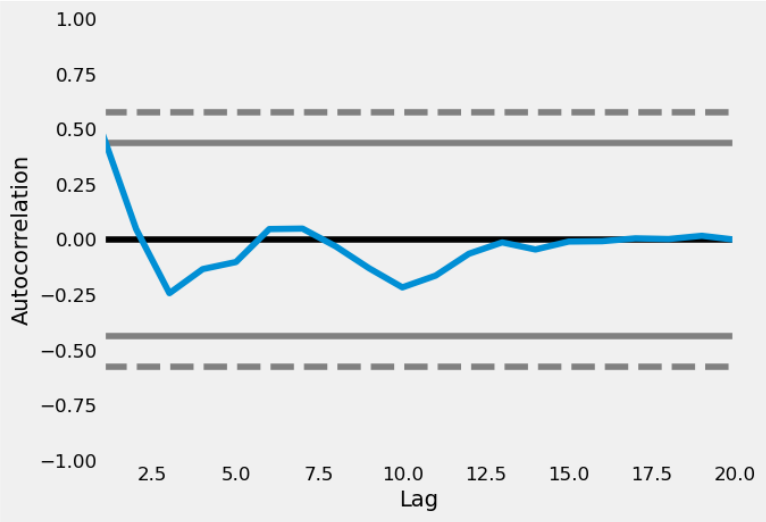


Image 7: Autocorrelation Plot 2022

Based on the insights gained from the autocorrelation plot and domain knowledge, the order parameters for the ARIMA model are chosen. This includes the order of differencing (d), the number of autoregressive terms (p), and the number of moving average terms (q). The time series data is split into training and testing sets using a split ratio of approximately 67:33. The ARIMA model is iteratively fitted to the training data using a loop. For each iteration, a new observation is added to the history of past observations, and a forecast is generated using the fitted model. The predicted values (predictions) are accumulated in the predictions list. The accuracy of the ARIMA model's forecasts is evaluated using the root mean squared error (RMSE). The mean_squared_error function from sklearn.metrics is employed to calculate the RMSE between the actual test data and the predicted values. The model was able to achieve a RMSE value of 75. The actual test data and the predicted values are plotted using matplotlib. This visual comparison allows an assessment of how well the ARIMA model aligns with the true values and provides insights into the model's forecasting performance.

In [50]:
```
# Fitting the data and predicting for time series model
# split into train and test sets
X = y.values
train, test = np.split(X, [int(.67 *len(X))])
history = [x for x in train]
predictions = list()
# walk-forward validation
for t in range(len(test)):
    model = ARIMA(history, order=(2,1,0))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
# evaluate forecasts
rmse = sqrt(mean_squared_error(test, predictions))
print('Test RMSE: %.3f' % rmse)
# plot forecasts against actual outcomes
pyplot.plot(test)
pyplot.plot(predictions, color='red')
pyplot.show()
```

```
Test RMSE: 74.879
```

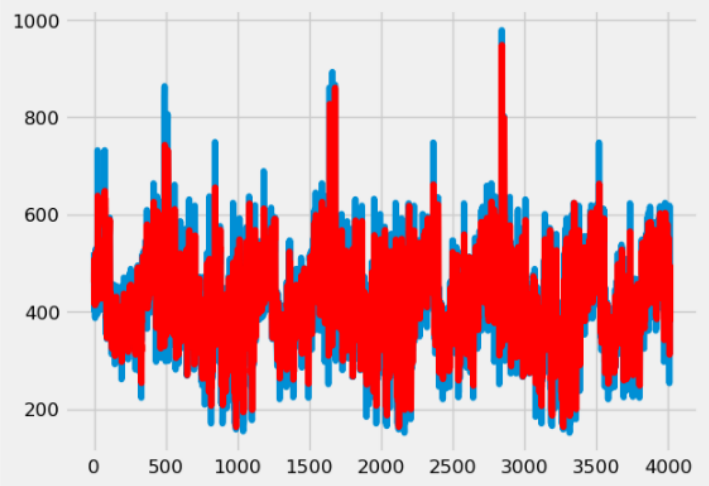


Image 8: Time Series Model; Predicted vs. Actual Plot

The graph above displays the predicted values(red) versus the actual values(blue) from the data set. The two are very close together, showing the accuracy of this prediction model. The plot_diagnostics method provided by the fitted ARIMA model is used to visualize its diagnostics. This includes assessing the distribution of residuals, checking for normality, and examining autocorrelation of residuals.

```
In [55]: model_fit.plot_diagnostics(figsize=(16, 8))
         plt.show()
```

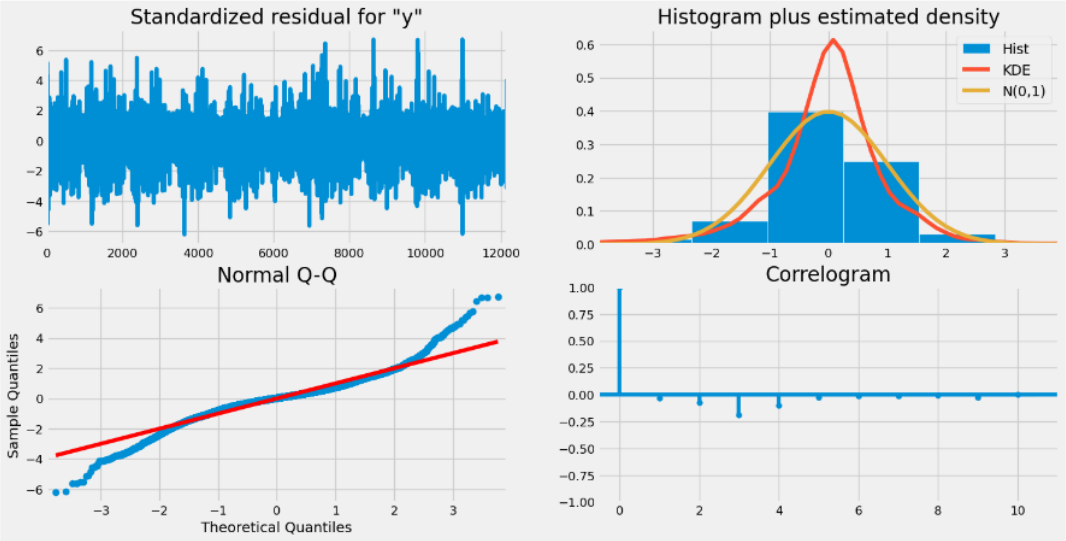


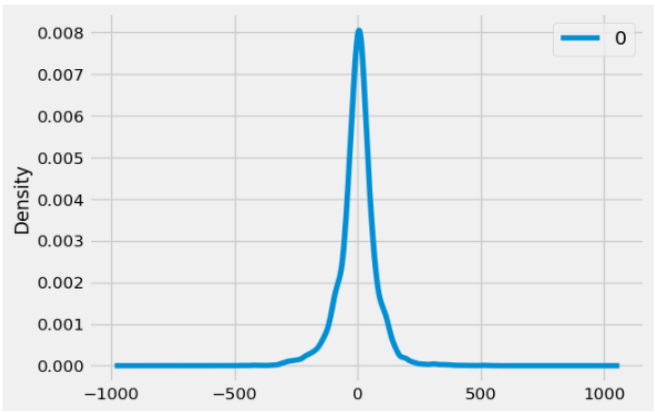
Image 9: Diagnostic Plots of Time Series Model

Additionally, line plots and density plots of residuals were created to identify any patterns or deviations from randomness. In the Normal Q-Q plot, it shows the expected distribution(red) compared to the predicted distribution(blue). In the plot, the predicted and expected are similar, showing that the predictions are relatively accurate.

```
In [56]: # line plot of residuals
         residuals = pd.DataFrame(model_fit.resid)
         residuals.plot()
         pyplot.show()
         # density plot of residuals
         residuals.plot(kind='kde')
         pyplot.show()
         # summary stats of residuals
         print(residuals.describe())
```

Image 10: Plotting Density

The results in the density plot below display the mean as 0.027, so as it is closer to 0, there is less bias in the model.




```
                 0
count  12160.000000
mean       0.027684
std       76.365601
min     -473.603742
25%      -33.112695
50%        3.632984
75%       36.913153
max      551.000000
```

Image 11: Density Plot

III. RESULTS/ANALYSIS

The results obtained from the data analysis and modeling process shed light on significant insights regarding the automotive industry's trajectory with respect to environmental sustainability and vehicle design optimization. This section discusses the key findings and implications derived from the conducted research. The correlation analysis revealed notable relationships between various attributes and CO2 emissions in automobiles. Among these, engine displacement, cylinder count, gear count, and air aspiration method exhibited correlations with CO2 emissions. These correlations align with the initial assumption that attributes such as turbocharging and regenerative braking would contribute to lower CO2 emissions and enhanced fuel efficiency. The linear regression model provided a predictive framework for CO2 emissions based on the selected features. The model achieved an impressive R2 score of approximately 86.6%. This score indicates that around 86.6% of the variance in CO2 emissions can be explained by the linear relationship with the chosen attributes. Notably, attributes such as engine displacement, cylinder count, and battery specifications played pivotal roles in the model's accuracy. The high R2 score demonstrates the efficacy of the linear regression model in capturing the relationships between features and CO2 emissions. The time series analysis using ARIMA modeling facilitated the identification of temporal patterns and trends in CO2 emissions data. The model achieved a Root Mean Squared Error (RMSE) of approximately 75. This RMSE value provides an indication of the average difference between the actual CO2 emissions and the model's predicted values. While the RMSE value is a bit higher than desired, it still shows the model's ability to capture and predict trends over time.

IV.DISCUSSION/CONCLUSION

In conclusion, this research project set out to provide a comprehensive forecast of the automotive industry's trajectory, focusing on advancements in environmental sustainability and vehicle design optimization. By employing data science techniques such as linear regression modeling and time series analysis, the study aimed to uncover insights that could guide manufacturers, policymakers, consumers, and other stakeholders toward a greener and more efficient automotive sector. The findings of this research contribute valuable insights to multiple dimensions of the automotive industry. Manufacturers can leverage the accurate forecasting capabilities of the linear regression model to anticipate trends and align their strategies with projected improvements. Policymakers have the opportunity to shape regulations and policies that accelerate the adoption of sustainable practices, thereby promoting a cleaner transportation ecosystem. Consumers are empowered to make informed decisions when purchasing vehicles, contributing to a greener future and potentially reducing operational costs. Furthermore, the research underscores the environmental impact of the automotive industry and how advancements in sustainability can lead to reduced emissions and resource consumption. Academic researchers and industry professionals can build upon these findings to further explore energy-efficient vehicle technologies and alternative fuels. From an economic perspective, stakeholders can make well-informed decisions and attract investments in areas that promise both growth and sustainability. In essence, this research serves as a roadmap for stakeholders to navigate the evolving landscape of the automotive industry. By fostering informed decision-making, strategic planning, and sustainable practices, the research

contributes to a more environmentally conscious and efficient automotive sector. As the industry continues to evolve, these insights will play a pivotal role in steering the course toward a greener and more sustainable future.

For the full project, see github link below.
Github

REFERENCES

Shatby, S. E. (2022, May 18). How to build a predictive model in python?. 365 Data Science. https://365datascience.com/tutorials/python-tutorials/predictive-model-python/
U.S Department of Energy. (n.d.). Download Fuel Economy Data. www.fueleconomy.gov - the official government source for fuel economy information. https://www.fueleconomy.gov/feg/download.shtml
Brownlee, J. (2020, December 9). How to create an Arima model for time series forecasting in Python. MachineLearningMastery.com. https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
Li, S. (2018, September 5). An end-to-end project on time series analysis and forecasting with python. Medium. https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b

ACKNOWLEDGMENTS

I would like to acknowledge my mentor Lori Chiu who made this project possible. Her help and guidance led through the coding stages of my project.