SALES ANALYSIS OF ELECTRIC VEHICLES
EV Sales Prediction in the World Using Machine Learning

Aim: To predict the sales of EV and its share in the transportation sector using ML by 2030

Team:

Sanjay R (23M0002)
Saurabh Narayan Pawar (23M0003)
Vanhar Ali Shaik (23M0007)
Ranjith A V (23M0009)

Course Instructor:

Prof. Amit Sethi

Course:

EE 769: Introduction to Machine Learning

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
Mumbai - 400076, India

April, 2024

ABSTRACT

The surge in Electric Vehicle (EV) sales is transforming the global transportation landscape, prompting a crucial need for accurate sales predictions to inform policy, industry strategies, and individual choices. This abstract outlines a predictive model leveraging Machine Learning (ML), particularly Linear Regression, to forecast EV sales and its share in the transportation sector worldwide by 2030.
The primary aim of this study is to develop a robust predictive model that anticipates the trajectory of EV sales globally, shedding light on its evolving role in the transportation sector. Specifically, the model targets predictions up to 2030, providing insights crucial for strategic planning, policy formulation, and investment decisions.

The dataset utilized for this analysis is sourced from the International Energy Agency (IEA), specifically the "Global EV Outlook 2022" dataset. This comprehensive dataset offers a rich repository of historical EV sales data, market trends, and regional insights, serving as a reliable foundation for the predictive model.

The chosen methodology revolves around employing Linear Regression, a fundamental ML algorithm, to model the relationship between various predictors such as historical sales data, economic indicators, policy changes, and technological advancements, and the target variable - EV sales. By training the model on historical data and validating it against known outcomes, the aim is to create a predictive tool capable of forecasting EV sales accurately.

The outcomes of this predictive model extend benefits to diverse stakeholders like, Common Citizens seeking to transition to EVs can make informed decisions based on predicted market trends and availability. EV Industry, i.e., Manufacturers, suppliers, and stakeholders within the EV ecosystem can utilize these predictions to optimize production, distribution, and investment strategies, fostering sustainable growth. Policymakers gain valuable insights into the potential impact of regulatory measures, incentives, and infrastructure investments on EV adoption, aiding in the formulation of effective policies to accelerate the transition to sustainable mobility.Investors in the EV sector can make data-driven investment decisions, mitigating risks and capitalizing on emerging opportunities in the rapidly evolving EV market.

In conclusion, this predictive model harnesses the power of ML to offer actionable insights into the future trajectory of EV sales worldwide, empowering stakeholders to navigate the transition towards sustainable transportation effectively.

INTRODUCTION

Electric vehicles (EVs) have emerged as a disruptive force in the automotive industry, representing a pivotal shift towards sustainable transportation. With advancements in battery technology, supportive government policies, and growing environmental awareness among consumers, the adoption of EVs has been steadily increasing worldwide.
Sales analysis of electric vehicles provides valuable insights into market trends, consumer preferences, and the future trajectory of the EV market. By examining historical sales data, analysts can identify patterns, forecast future demand, and assess the impact of various factors on EV sales.

From a global perspective, regions such as China, Europe, and North America have emerged as key markets for electric vehicles, with each region experiencing varying degrees of adoption driven by local incentives, infrastructure development, and regulatory frameworks.
Moreover, the rise of electric vehicles extends beyond passenger cars to encompass a diverse range of vehicles, including electric buses, trucks, and two-wheelers. This diversification of the EV market underscores its potential to revolutionize not only personal transportation but also public transit and freight logistics.
As stakeholders seek to understand the dynamics of the electric vehicle market, data-driven approaches such as machine learning (ML) have gained prominence. By leveraging ML algorithms to analyze large datasets comprising sales figures, demographic information, and socio-economic factors, researchers can develop predictive models to forecast EV sales and assess their share in the transportation sector by 2030.

Incorporating machine learning into sales analysis enables stakeholders to anticipate market trends, optimize production strategies, and formulate effective policy interventions to accelerate the transition towards sustainable mobility. By harnessing the power of data and analytics, the journey towards a greener, more efficient transportation ecosystem driven by electric vehicles becomes increasingly feasible and impactful.
This introduction sets the stage for your analysis by providing context on the significance of sales analysis in understanding the EV market dynamics and introduces the role of machine learning in predicting future sales trends and EV's share in the transportation sector.

METHODOLOGY

The code imports necessary libraries for data analysis and visualization:

numpy as np: NumPy is imported for numerical computing tasks.
pandas as pd: Pandas is imported for data manipulation and analysis.
matplotlib.pyplot as plt: Matplotlib is imported for creating visualizations.
seaborn as sns: Seaborn is imported for statistical data visualization.
geopandas as gpd: Geopandas is imported, possibly for geographical data visualization or analysis.
plotly.express as px: Plotly Express is imported for interactive visualization.
plotly.graph_objects as go: Plotly Graph Objects is imported, likely for more advanced plot customization.

df = pd.read_csv("IEA-EV-data.csv")

This line reads a CSV (Comma Separated Values) file named "IEA-EV-data.csv" and loads its contents into a Pandas DataFrame named df. The pd.read_csv() function from the Pandas library is used to read CSV files into DataFrames.

df.head(50)

This line displays the first 50 rows of the DataFrame df. The head() method is used to retrieve the first few rows (by default, 5 rows) of the DataFrame, and specifying 50 as an argument displays the first 50 rows.

df.shape

This line retrieves the dimensions of the DataFrame df, specifically the number of rows and columns. The shape attribute of a DataFrame returns a tuple containing the number of rows followed by the number of columns.

df.info()

This line provides a concise summary of the DataFrame df, including information such as the data types of each column, the number of non-null values in each column, memory usage, and the index dtype. The info() method of a DataFrame is used for this purpose.

df.nunique()

This line calculates the number of unique values in each column of the DataFrame df. The nunique() method returns a Series with the count of unique values for each column.

df['unit'].unique()

This line retrieves an array of unique values from the column named 'unit' in the DataFrame df. It filters the 'unit' column and returns an array containing only the unique values present in that column.

sales = df.loc[lambda df: df['unit'] == 'sales']

This line filters rows in the DataFrame df where the value in the 'unit' column is 'sales'. The loc[] method is used to locate rows based on a condition specified inside a lambda function. The result is stored in a new DataFrame named sales.

sales.head(50)

This line displays the first 50 rows of the DataFrame sales. It allows us to inspect the filtered DataFrame to see if the filtering operation was successful.

df['year'].unique()

This line retrieves an array of unique values from the 'year' column in the DataFrame df. It provides information about the distinct years present in the dataset, which could be useful for further analysis.

fig = plt.figure(figsize=(6, 6))

This line creates a new figure for plotting with a specified size of 6x6 inches. The plt.figure() function from Matplotlib is used to create a new figure object.

ax = fig.subplots()

This line creates subplots on the figure fig. Since no arguments are provided to the subplots() method, it creates a single subplot. The subplot object is stored in the variable ax.

df.region.value_counts().head(5).plot(ax=ax, kind='pie', autopct='%1.1f%%', pctdistance=0.7)

This line generates a pie chart showing the distribution of the top 5 regions producing electric vehicles. It first calculates the value counts for each unique region in the 'region' column, then selects the top 5 values using head(5). The plot() function is used to create the pie chart, and various parameters are provided to customize the plot, such as the subplot (ax), chart type (kind='pie'), formatting of autopct (percentage display), and pctdistance (distance of percentage labels from the center of the pie).

ax.set_ylabel("")

This line sets the y-axis label of the plot to an empty string. It removes the default label associated with the y-axis.

ax.set_title("Leading Electric Vehicles Producers - Top 5")

This line sets the title of the plot to "Leading Electric Vehicles Producers - Top 5". It provides a descriptive title for the pie chart.

plt.show()

This line displays the plot. It is necessary to visualize the generated pie chart in the output.

sales['parameter'].unique()

This line retrieves an array of unique values from the 'parameter' column in the DataFrame sales. It provides information about the distinct parameters present in the dataset, which might include different aspects or categories related to EV sales.

parameter_year = sales.groupby(['year', 'parameter'])['value'].sum()

This line groups the sales DataFrame by 'year' and 'parameter', then calculates the sum of the 'value' column for each group. It effectively aggregates the sales data based on different parameters over the years.

parameter_year.to_csv('parameter_year.csv')

This line exports the grouped data to a CSV file named 'parameter_year.csv'. It saves the aggregated sales data grouped by year and parameter to a file for further analysis or reference.

parameter_year = pd.read_csv('parameter_year.csv')

This line reads the exported CSV file ('parameter_year.csv') containing the aggregated sales data back into a new DataFrame named parameter_year. It allows for further manipulation and visualization of the data.

parameter_year = parameter_year.pivot_table(parameter_year, index=['year'], columns=['parameter'], fill_value=0, aggfunc=np.sum)

This line creates a pivot table from the parameter_year DataFrame to reshape the data. It reorganizes the DataFrame so that the years are the index and the parameters become columns. The fill_value parameter is used to fill missing values with 0, and aggfunc=np.sum specifies that the sum function should be applied in case of duplicate entries.

parameter_year.plot(kind='bar', stacked=True, cmap='Set2')

This line creates a stacked bar chart of the parameter_year DataFrame. The bars represent different parameters, stacked on top of each other for each year. The kind='bar' parameter specifies the type of plot, while stacked=True stacks the bars. The cmap='Set2' parameter sets the color map for the plot.

plt.legend(['EV sales'])

This line adds a legend to the plot, labeling the stacked bars as 'EV sales'. It provides a visual guide to interpret the different sections of the stacked bars.

plt.xticks(rotation=45)

This line rotates the x-axis labels by 45 degrees for better readability. It prevents overlapping of labels when there are many data points on the x-axis.

plt.ylabel('Sales value')

This line adds a label to the y-axis of the plot, indicating that the values on the y-axis represent sales values.

plt.title('Sales Trend and Prediction of EV Sales: 2010 - 2030')

This line sets the title of the plot to 'Sales Trend and Prediction of EV Sales: 2010 - 2030'. It provides a descriptive title for the plot, specifying the timeframe covered by the analysis.

plt.show()

This line displays the plot. It is necessary to visualize the stacked bar chart in the output.

region_year = sales.groupby(['year', 'region'])['value'].sum()

This line groups the sales DataFrame by both 'year' and 'region', and then sums up the 'value' column for each group. This effectively aggregates the sales data based on both year and region.

```
region_year.to_csv('region_year.csv')
```

This line exports the grouped data to a CSV file named 'region_year.csv'. It saves the aggregated sales data grouped by year and region to a file for further analysis or reference.

```
region_year = pd.read_csv('region_year.csv')
```

This line reads the exported CSV file ('region_year.csv') containing the aggregated sales data back into a new DataFrame named region_year. It allows for further manipulation and visualization of the data.

```
region_year = region_year.pivot_table(region_year, index=['year'], columns=['region'], fill_value=0, aggfunc=np.sum)
```

This line creates a pivot table from the region_year DataFrame to reshape the data. It reorganizes the DataFrame so that the years are the index and the regions become columns. The fill_value parameter is used to fill missing values with 0, and aggfunc=np.sum specifies that the sum function should be applied in case of duplicate entries.

```
ax = region_year.plot(kind='bar', stacked=True)
```

This line creates a stacked bar chart of the region_year DataFrame. The bars represent different regions, stacked on top of each other for each year. The kind='bar' parameter specifies the type of plot, while stacked=True stacks the bars.

```
sns.move_legend(ax, loc='center right', bbox_to_anchor=(1.4, 0.5), title='Region', fontsize=7)
```

This line moves the legend of the plot to a specified position. The move_legend() function from Seaborn library is used to adjust the legend. The loc='center right' parameter specifies the location of the legend, and bbox_to_anchor=(1.4, 0.5) adjusts the anchor point to position the legend slightly outside the plot. The title='Region' parameter sets the title of the legend, and fontsize=7 specifies the font size.

```
plt.xticks(rotation=45)
```

This line rotates the x-axis labels by 45 degrees for better readability. It prevents overlapping of labels when there are many data points on the x-axis.

```
plt.ylabel('Sales value')
```

This line adds a label to the y-axis of the plot, indicating that the values on the y-axis represent sales values.

```
plt.title('Sales Trend and Prediction of Each Region: 2010 - 2030')
```

This line sets the title of the plot to 'Sales Trend and Prediction of Each Region: 2010 - 2030'. It provides a descriptive title for the plot, specifying the timeframe covered by the analysis.

```
plt.show()
```

This line displays the plot. It is necessary to visualize the stacked bar chart in the output.

```
sales['category'].unique()
```

This line retrieves an array of unique values from the 'category' column in the DataFrame sales. It provides information about the distinct categories of electric vehicles present in the dataset.

category_year = sales.groupby(['year', 'category'])['value'].sum()

This line groups the sales DataFrame by both 'year' and 'category', and then calculates the sum of the 'value' column for each group. It effectively aggregates the sales data based on both year and category.

category_year.to_csv('category_year.csv')

This line exports the grouped data to a CSV file named 'category_year.csv'. It saves the aggregated sales data grouped by year and category to a file for further analysis or reference.

category_year = pd.read_csv('category_year.csv')

This line reads the exported CSV file ('category_year.csv') containing the aggregated sales data back into a new DataFrame named category_year. It allows for further manipulation and visualization of the data.

category_year = category_year.pivot_table(category_year, index=['year'], columns=['category'], fill_value=0, aggfunc=np.sum)

This line creates a pivot table from the category_year DataFrame to reshape the data. It reorganizes the DataFrame so that the years are the index and the categories become columns. The fill_value parameter is used to fill missing values with 0, and aggfunc=np.sum specifies that the sum function should be applied in case of duplicate entries.

ax1 = category_year.plot(kind='line', cmap='Set2', marker='D')

This line creates a line chart of the category_year DataFrame. The lines represent different categories of electric vehicles plotted against the years. The kind='line' parameter specifies the type of plot, while cmap='Set2' sets the color map for the lines, and marker='D' specifies diamond markers for data points along the lines.

ax1.xaxis.set_major_formatter(plt.FormatStrFormatter('%.0f'))

This line formats the x-axis tick labels to display integers without decimal places. It ensures that the years are displayed as whole numbers on the x-axis.

plt.legend(['Historical', 'Projection-APS', 'Projection-STEPS'])

This line adds a legend to the plot, labeling the lines as 'Historical', 'Projection-APS', and 'Projection-STEPS'. It provides a visual guide to interpret the different lines on the plot.

plt.xticks(rotation=45)

This line rotates the x-axis labels by 45 degrees for better readability. It prevents overlapping of labels when there are many data points on the x-axis.

plt.ylabel('Sales value')

This line adds a label to the y-axis of the plot, indicating that the values on the y-axis represent sales values.

plt.title('Sales Trend and Prediction of Category: 2010 - 2030')

This line sets the title of the plot to 'Sales Trend and Prediction of Category: 2010 - 2030'. It provides a descriptive title for the plot, specifying the timeframe covered by the analysis.

plt.show()

This line displays the plot. It is necessary to visualize the line chart in the output.

mode_year = sales.groupby(['year', 'mode'])['value'].sum()

This line groups the sales DataFrame by both 'year' and 'mode', and then calculates the sum of the 'value' column for each group. It effectively aggregates the sales data based on both year and mode.

mode_year.to_csv('mode_year.csv')

This line exports the grouped data to a CSV file named 'mode_year.csv'. It saves the aggregated sales data grouped by year and mode to a file for further analysis or reference.

mode_year = pd.read_csv('mode_year.csv')

This line reads the exported CSV file ('mode_year.csv') containing the aggregated sales data back into a new DataFrame named mode_year. It allows for further manipulation and visualization of the data.

mode_year = mode_year.pivot_table(mode_year, index=['year'], columns=['mode'], fill_value=0, aggfunc=np.sum)

This line creates a pivot table from the mode_year DataFrame to reshape the data. It reorganizes the DataFrame so that the years are the index and the different modes become columns. The fill_value parameter is used to fill missing values with 0, and aggfunc=np.sum specifies that the sum function should be applied in case of duplicate entries.

ax2 = mode_year.plot(kind='line', cmap='Set2', marker='D')

This line creates a line chart of the mode_year DataFrame. The lines represent different modes of electric vehicles plotted against the years. The kind='line' parameter specifies the type of plot, while cmap='Set2' sets the color map for the lines, and marker='D' specifies diamond markers for data points along the lines.

ax2.xaxis.set_major_formatter(plt.FormatStrFormatter('%.0f'))

This line formats the x-axis tick labels to display integers without decimal places. It ensures that the years are displayed as whole numbers on the x-axis.

plt.legend(['Buses', 'Cars', 'Trucks', 'Vans'])

This line adds a legend to the plot, labeling the lines as 'Buses', 'Cars', 'Trucks', and 'Vans'. It provides a visual guide to interpret the different lines on the plot.

plt.xticks(rotation=45)

This line rotates the x-axis labels by 45 degrees for better readability. It prevents overlapping of labels when there are many data points on the x-axis.

plt.ylabel('Sales value')

This line adds a label to the y-axis of the plot, indicating that the values on the y-axis represent sales values.

```
plt.title('Sales Trend and Prediction of Mode: 2010 - 2030')
```

This line sets the title of the plot to 'Sales Trend and Prediction of Mode: 2010 - 2030'. It provides a descriptive title for the plot, specifying the timeframe covered by the analysis.

```
plt.show()
```

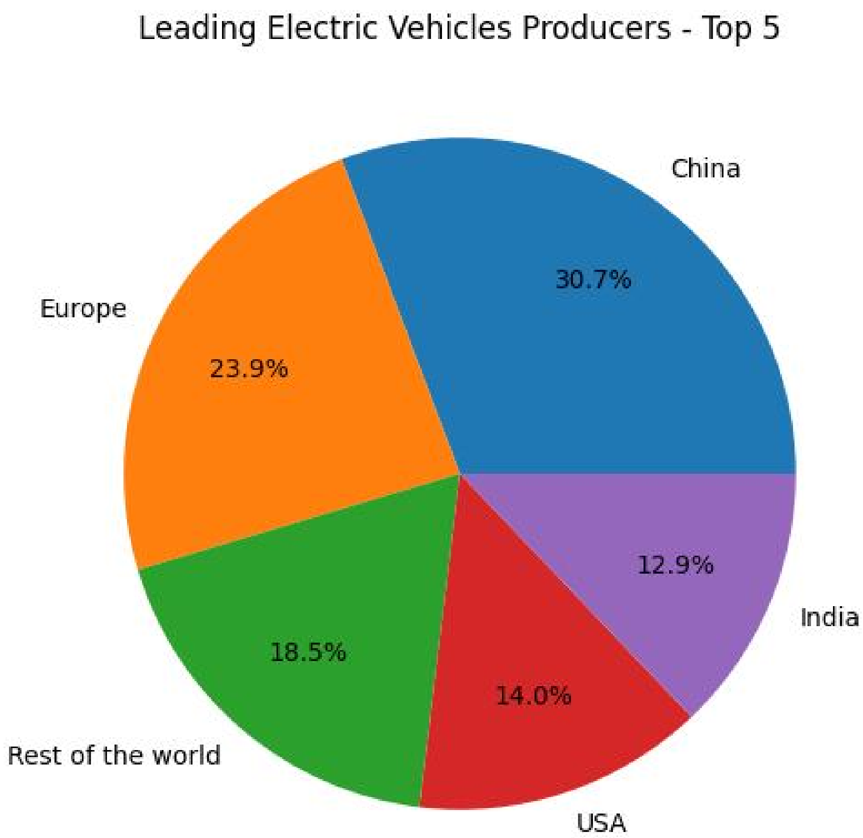
This line displays the plot. It is necessary to visualize the line chart in the output.

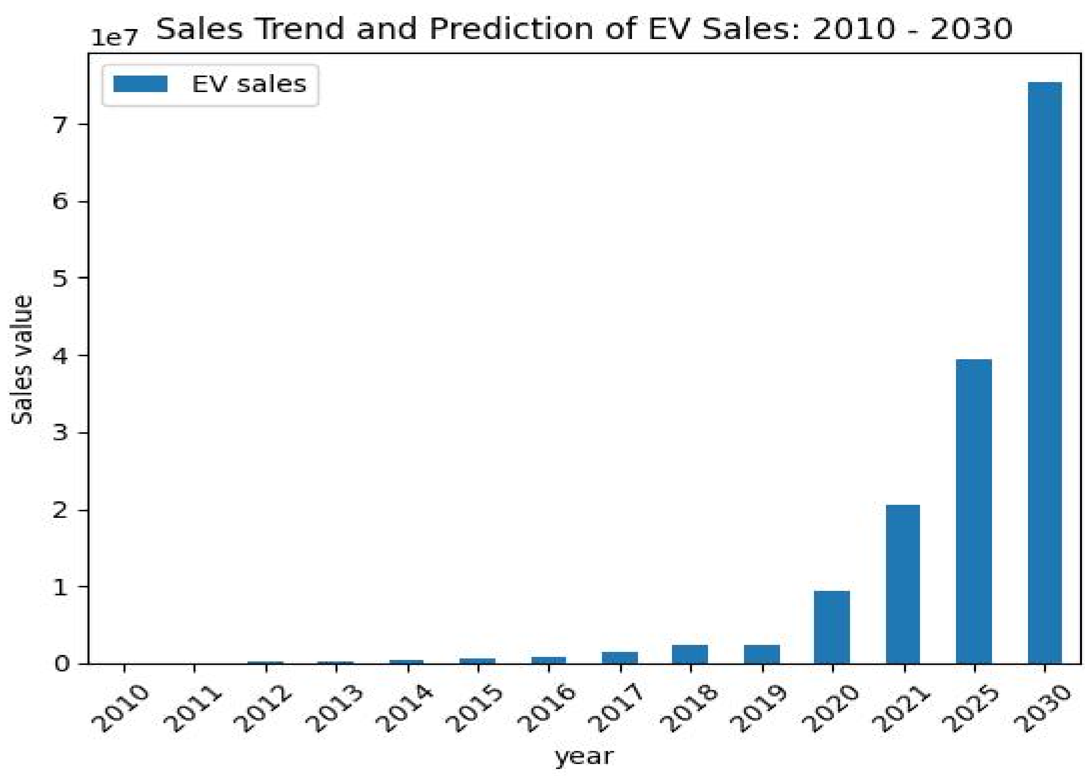
```
sales['powertrain'].unique()
```

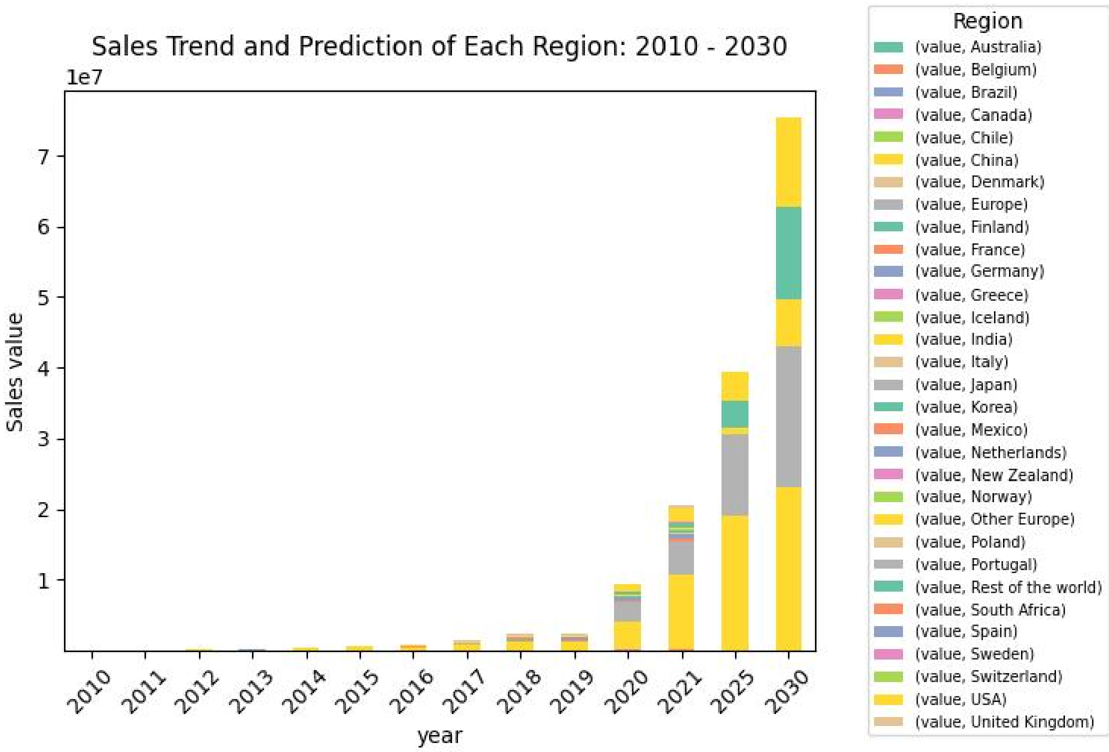
This line retrieves an array of unique values from the 'powertrain' column in the DataFrame sales. It provides information about the different powertrain types present in the dataset.

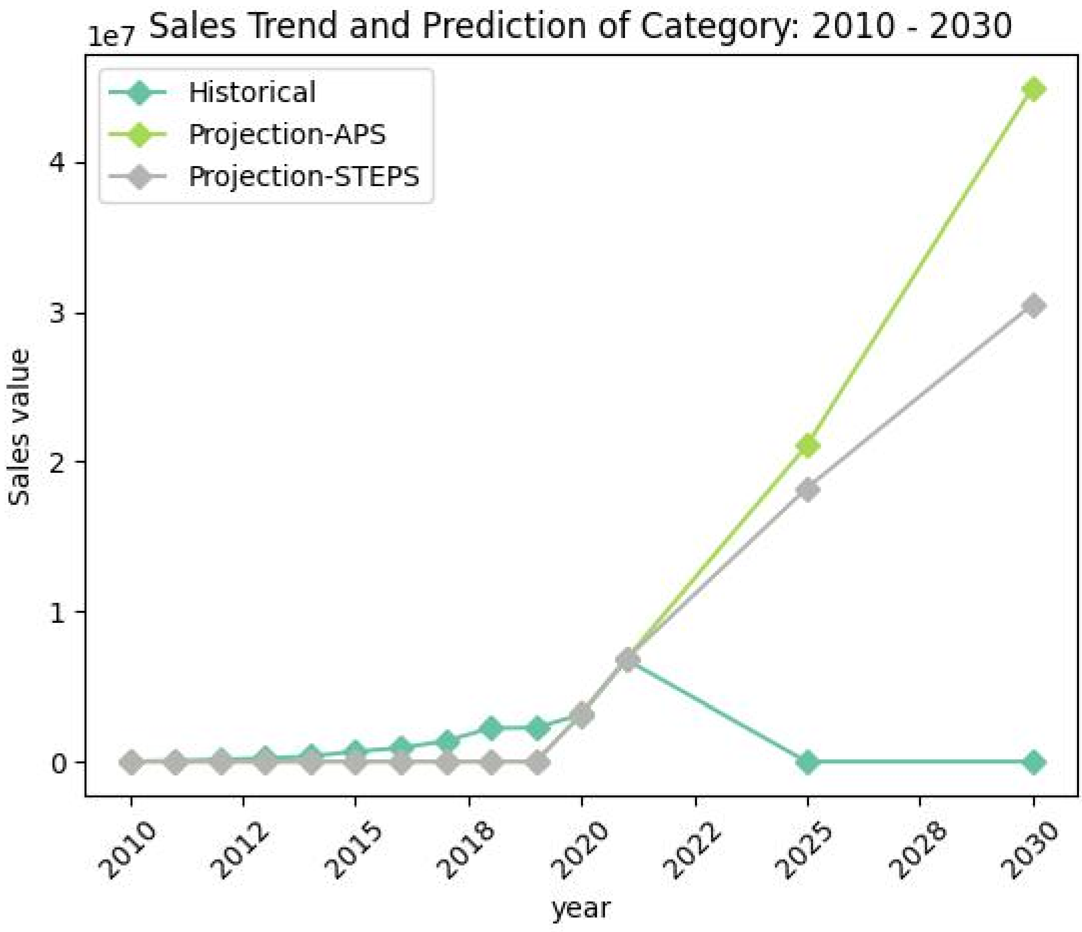
Similar operations to the ones above, for the 'powertrain' column are followed. They group the data by year and powertrain type, aggregate the sales values, export the data to a CSV file, read the CSV file back into a DataFrame, create a pivot table, and finally plot a line chart showing the sales trend and prediction for each powertrain type over the specified timeframe.

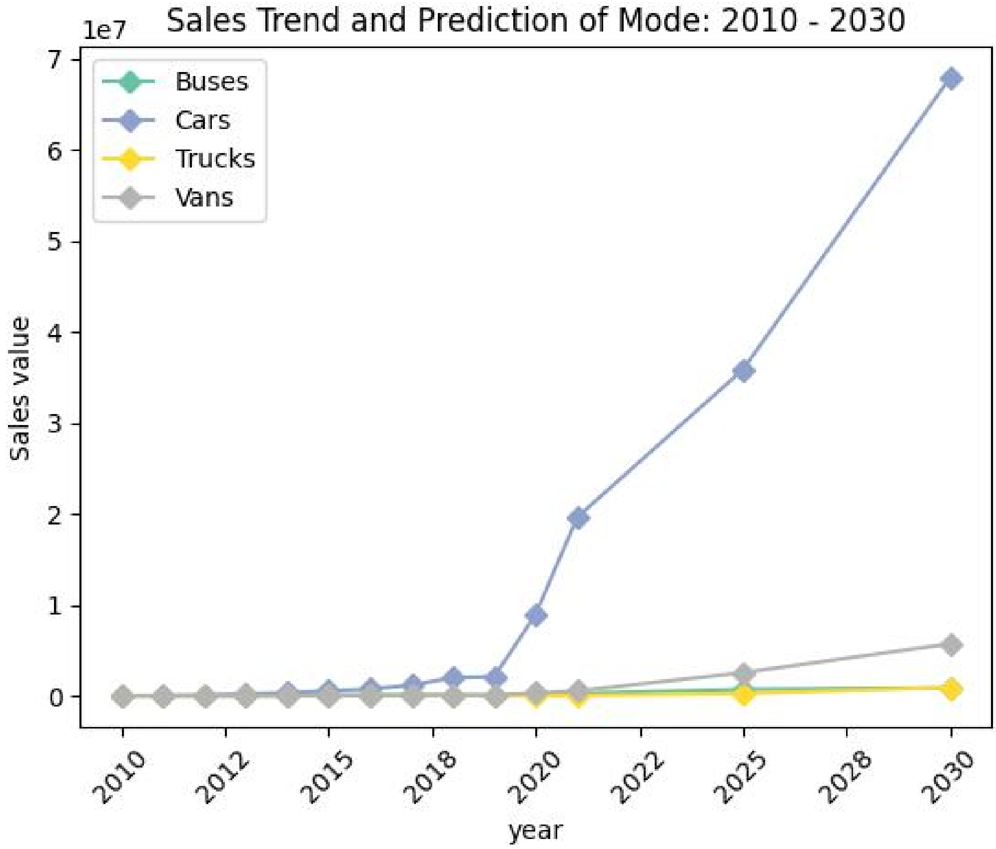
RESULTS

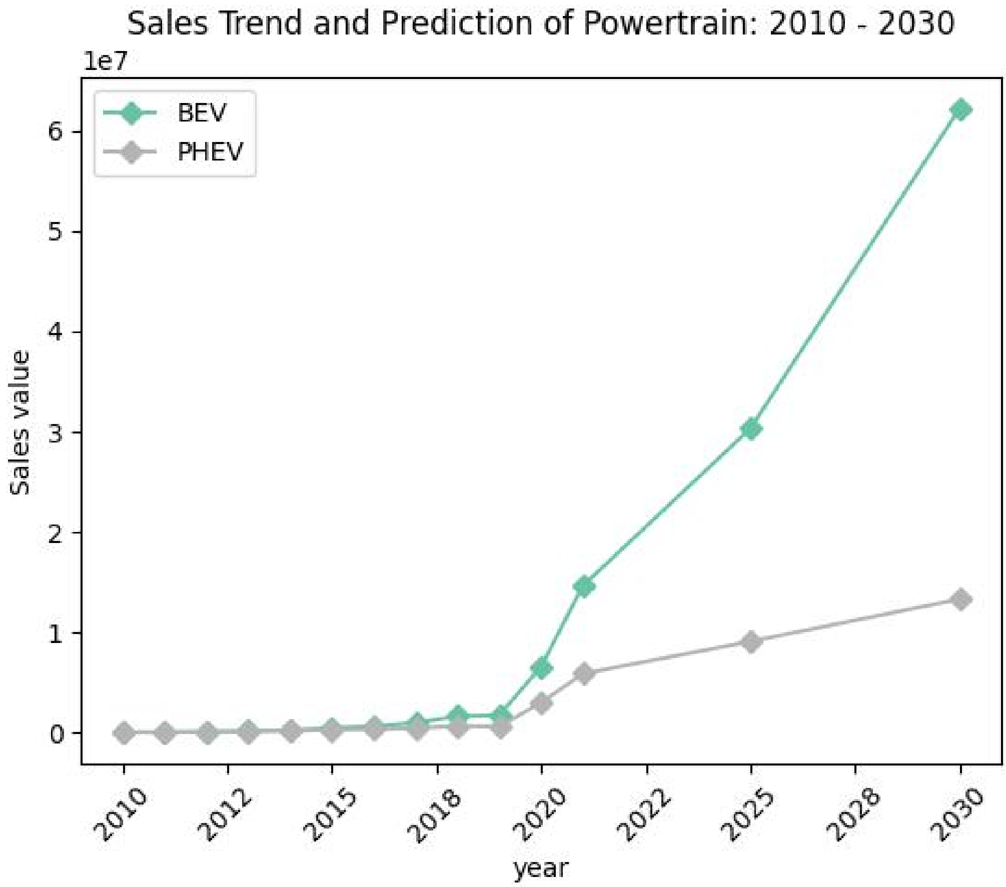

Leading Electric Vehicles Producers - Top 5
China
30.7%
Europe
23.9%
Rest of the world
18.5%
USA
14.0%
India
12.9%


Sales Trend and Prediction of EV Sales: 2010 - 2030
1e7
EV sales
Sales value
0
1
2
3
4
5
6
7
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2025
2030
year


Sales Trend and Prediction of Each Region: 2010 - 2030
1e7
Sales value
year
Region
(value, Australia)
(value, Belgium)
(value, Brazil)
(value, Canada)
(value, Chile)
(value, China)
(value, Denmark)
(value, Europe)
(value, Finland)
(value, France)
(value, Germany)
(value, Greece)
(value, Iceland)
(value, India)
(value, Italy)
(value, Japan)
(value, Korea)
(value, Mexico)
(value, Netherlands)
(value, New Zealand)
(value, Norway)
(value, Other Europe)
(value, Poland)
(value, Portugal)
(value, Rest of the world)
(value, South Africa)
(value, Spain)
(value, Sweden)
(value, Switzerland)
(value, USA)
(value, United Kingdom)


Sales Trend and Prediction of Category: 2010 - 2030
1e7
Historical
Projection-APS
Projection-STEPS
Sales value
year


Sales Trend and Prediction of Mode: 2010 - 2030
1e7
7
6
5
4
3
2
1
0
Buses
Cars
Trucks
Vans
Sales value
2010
2012
2015
2018
2020
2022
2025
2028
2030
year


Sales Trend and Prediction of Powertrain: 2010 - 2030
1e7
6
5
4
3
2
1
0
BEV
PHEV
Sales value
2010
2012
2015
2018
2020
2022
2025
2028
2030
year

CONCLUSION

1. Dramatic Increase in EV Sales: The statement highlights a significant trend in the electric vehicle market, indicating a substantial surge in sales over recent years, particularly post-2020. This increase likely stems from several factors, including advancements in EV technology, growing environmental awareness, governmental incentives promoting EV adoption, and improvements in charging infrastructure. As concerns about climate change intensify and consumers seek alternatives to traditional fossil fuel-powered vehicles, the demand for EVs has witnessed a remarkable uptick.

2. China's Leadership in EV Sales: China's dominance in the global EV market is underscored by its leading position in terms of sales volume. This leadership position can be attributed to China's proactive approach towards promoting electric mobility, characterized by robust government policies incentivizing EV adoption, substantial investments in research and development, and a burgeoning domestic EV industry. Japan and India's close proximity in EV sales indicates a regional shift towards embracing electric mobility, driven by similar factors such as environmental concerns, policy support, and technological advancements.

3. Rise of EV Sales in India: The acknowledgment of India's increasing EV sales reflects a noteworthy trend within the country's automotive sector. India, spurred by its ambitious electric mobility goals and initiatives, is witnessing a gradual but discernible rise in EV adoption. Factors such as government subsidies, tax incentives, investments in charging infrastructure, and the introduction of affordable EV models are contributing to this upward trajectory in EV sales.

4. Projected Growth in EV Adoption: The projection of steady EV adoption in nations including Belgium, Greece, China, the United States, and India signifies a global shift towards sustainable transportation. This trend is fueled by a combination of factors, including regulatory measures aimed at reducing greenhouse gas emissions, technological advancements making EVs more accessible and affordable, and increasing consumer awareness about the benefits of electric mobility. The projected growth underscores the transformative potential of EVs in mitigating environmental concerns and reducing dependence on fossil fuels.

5. Popularity of EV Vehicle Categories: The statement emphasizes the dominance of cars as the most popular EV vehicle category, while also noting a significant increase in sales for EV vans, buses, and trucks. This diversification in the EV market reflects a growing demand for electric vehicles across various segments of the transportation sector, including personal, commercial, and public transport. The expansion of EV offerings beyond traditional passenger cars highlights the versatility and adaptability of electric mobility solutions to meet diverse transportation needs.

6. BEV vs. PHEV Growth: The comparison between Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) highlights the remarkable growth trajectory of BEVs. This trend can be attributed to several factors, including advancements in battery technology, expanded charging infrastructure, lower operating costs, and consumer preferences for pure electric vehicles. The phenomenal growth of BEVs underscores the shift towards all-electric transportation solutions, driven by concerns about air quality, energy security, and climate change.

7. Consideration of External Factors: The acknowledgment of potential external influences impacting EV sales underscores the complexity of predicting future market trends accurately. External factors such as regulatory changes, fluctuations in energy prices, technological breakthroughs, geopolitical developments, and unforeseen events like pandemics can significantly impact the trajectory of EV sales. While the presented projections offer valuable insights, they are subject to revision based on evolving market dynamics and external factors beyond the model's scope.

REFERENCES

1. Jeannie Yeh, Yu-Ting Wang, A Prediction Model for Electric Vehicle Sales Using Machine Learning Approaches, January 2023, Journal of Global Information Management 31(1):1-21. Follow journal DOI: 10.4018/JGIM.327277

2. Yang, A., Liu, C., Yang, D., & Lu, C. (2023). Electric vehicle adoption in a mature market: A case study of Norway. Journal of Transport Geography, 106, 103489. doi:10.1016/j.jtrangeo.2022.103489

3. Requia, W. J., Mohamed, M., Higgins, C. D., Arain, A., & Ferguson, M. (2018). How clean are electric vehicles? Evidence-based review of the effects of electric mobility on air pollutants, greenhouse gas emissions and human health. Atmospheric Environment, 185, 64–77. doi:10.1016/j.atmosenv.2018.04.040

4. https://www.iea.org/data-and-statistics/data-product/global-ev-outlook-2022