Analyzing and Visualizing the EV Vehicles Population.

Praveen Kumar Chidipothu[1] Praveen Babu Narni[2] Ramakrishna Reddy Boggula[3] Trinadh Machanavazzala[4]

Northwest Missouri State University, Maryville MO 64468, USA
{S562112, s562887,S559309,S559254}@nwmissouri.edu

Abstract. The global automotive industry is embracing electric vehicles (EVs) as a solution to environmental and energy concerns. This study analyzes EV population data from diverse sources to offer a comprehensive understanding of the market. Through advanced visualization techniques, including interactive maps and charts, it reveals insights into adoption rates, geographical distribution, and influencing factors. By providing actionable insights for policymakers and industry stakeholders, this research contributes to accelerating the transition to sustainable transportation. In conclusion, it serves as a valuable resource for decision-makers navigating the complexities of the EV market, driving towards a greener future.

1 Introduction

The global automotive industry is witnessing a profound shift towards electric vehicles (EVs), driven by escalating concerns about environmental sustainability and energy efficiency. This study embarks on a comprehensive analysis and visualization of the EV vehicle population, leveraging cutting-edge tools and technologies. Utilizing Pyspark for efficient data preprocessing and Tableau for advanced visualization, this research draws upon diverse data sources including governmental agencies, industry reports, and academic studies. Through the examination of key metrics such as geographical distribution, adoption rates, vehicle types, and technological advancements, the study aims to unravel the evolving dynamics of the EV ecosystem.

By harnessing the capabilities of Pyspark and Tableau, this research endeavors to provide stakeholders with a sophisticated understanding of the EV market landscape. These tools facilitate seamless data access, visualization, and analysis, enabling stakeholders to derive actionable insights. Additionally, by exploring factors influencing EV adoption – encompassing policy incentives, infrastructure development, consumer preferences, and economic considerations – this study aims to guide decision-makers towards fostering a transition to a greener, more sustainable transportation system. Thus, through the integration of Pyspark and Tableau, this research contributes significantly to the ongoing discourse on EVs,

offering a comprehensive toolkit for navigating the complexities of the market and driving towards a more sustainable future.

1. Data Acquisition: Download the EV population dataset from Kaggle in CSV format.
2. Data Preprocessing: Use Python libraries such as Pandas to read the CSV file and perform necessary data preprocessing tasks like handling missing values, data cleaning, and formatting.
3. Visualization with Tableau: Utilize Tableau, a popular plotting library in Python, to create various visualizations such as line plots, bar charts, and histograms to represent different aspects of the EV population data. For instance:
   a. Line plots to show the trend of EV adoption over time.
   b. Bar charts to compare the EV population across different regions or manufacturers.
   c. Histograms to visualize the distribution of EVs by model year.

2 Architecture

The architecture for this project involves a structured approach to data gathering, preprocessing, analysis, and visualization, leveraging key technologies to achieve the research goals efficiently and effectively.

1. Data Gathering: The primary data source for this project is the EV population dataset obtained from Kaggle in CSV format. This dataset contains comprehensive information on EV adoption rates, vehicle types, geographical distribution, and other relevant metrics.
2. Data Preprocessing: The initial step involves extracting the data from the CSV file and loading it into a data frame. Subsequently, data preprocessing is performed to clean the dataset and remove any unrelated data using SQL queries. Pyspark is utilized for this purpose, leveraging its powerful capabilities for large-scale data processing and manipulation.
3. Analysis: Once the data is cleaned and prepared, it is ready for analysis to address the research goals outlined, including comparing EV adoption rates across regions, examining types of electric vehicles, analyzing adoption trends by model year, and exploring price sensitivity. These analyses are conducted using appropriate statistical methods and algorithms implemented in Python.
4. Visualization: The preprocessed data is then visualized using Tableau, a powerful data visualization tool. Various types of plots and graphs, such as bar charts, line plots, and histograms, are created to represent different aspects of the EV population data. These visualizations facilitate the interpretation of complex datasets and enable stakeholders to identify patterns, trends, and outliers within the data.
5. Comparison and Presentation: Finally, the visualizations are compared and integrated into a comprehensive presentation, allowing for a holistic view of

the EV market landscape. Insights derived from the analysis and visualization process are communicated effectively to stakeholders, including policymakers, industry stakeholders, and researchers, to inform decision-making and support the transition towards a sustainable transportation system.
Overall, this architecture ensures a systematic and data-driven approach to analyzing and visualizing the EV vehicle population, aligning with research goals and leveraging cutting-edge tools and technologies for maximum efficiency and effectiveness.

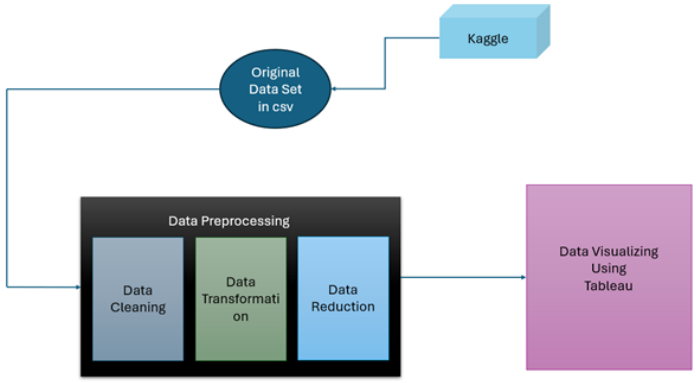


Fig. 1. High-level Architecture Of Data flow

3 Project Description

Consider including metrics (data quality, 5Vs, latency, processing time, resource utilization, security, cost) as project requirement

1. Data Quality: PySpark's DataFrame API allows for data quality checks, including missing values, consistency, and handling outliers. Validating data integrity prior to analysis ensures consistent results and decision-making.
   Using PySpark's capabilities in Jupyter Notebook allows for interactive exploration of data quality issues as well as data cleansing tasks that can be completed directly in your development environment.
2. 5V's Volume: PySpark's distributed processing allows for the efficient handling of large amounts of data. As the EV population dataset grows over time, PySpark's scalability ensures that processing continues without performance degradation.
   Variety: The dataset may contain a variety of data types, including numerical, categorical, and temporal. PySpark's DataFrame API supports a variety

of data formats and structures, allowing for the analysis of a wide range of datasets.
Velocity: PySpark's ability to process data in real-time or near-real-time allows for timely analysis of streaming data, such as updates to the EV population dataset. Quick iterations in Jupyter Notebook enable rapid prototyping and experimentation.
Value: Your analysis provides value to stakeholders by extracting insights from the EV population dataset, such as adoption trends, geographical patterns, and vehicle preferences, allowing for more informed decision-making.

3. Latency:
   PySpark's in-memory processing and parallel execution reduce latency for data processing tasks, allowing for quick analysis of the EV population dataset. The interactive environment of Jupyter Notebook allows for real-time data exploration and analysis.
   Tableau's integration with PySpark enables interactive visualizations with low latency, allowing stakeholders to explore insights dynamically.

4. Processing Time: PySpark code is optimized for efficiency, and distributed computing capabilities are used to reduce processing time. Techniques like caching intermediate results and employing appropriate partitioning strategies improve performance.
   Jupyter Notebook's integration with PySpark allows for faster development and testing of data processing pipelines, reducing iteration cycles.
   Tableau's visualization capabilities allow stakeholders to quickly interpret analysis results, which improves decision-making efficiency.
5. Resource Utilization: PySpark dynamically allocates cluster resources based on workload demands, resulting in optimal resource utilization. Monitor cluster metrics and adjust configuration settings to ensure efficient resource allocation.
   Integrating Tableau with PySpark allows you to share visualizations with multiple users while effectively managing server resources to handle varying workloads.
6. Security: Securing data in transit and at rest is essential. Implementing encryption, access controls, and authentication mechanisms ensures data security and compliance.
   Integrating authentication mechanisms between PySpark, Jupyter Notebook, and Tableau ensures secure access to sensitive data and analysis results.
7. Cost: Cost considerations include infrastructure costs for hosting PySpark clusters, Tableau licensing fees, and maintenance costs. Optimizing resource utilization and workload management helps to reduce operational costs while increasing the value derived from analytics efforts.

4 Research Goals

1. Considering 5 V's into the picture displaying all the data

2. Volume (EV Adoption Rates Across Regions): Compare EV adoption rates across different regions to understand the volume of EV adoption in each geographic area, providing insights into regional preferences and trends.
3. Volume (Total Count of EV Vehicle Sales): Determine the total count of EV vehicle sales to date, providing a quantitative measure of the volume of EV sales and tracking the growth trajectory of the EV market.
4. Variety (Examination of Electric Vehicle Types): Examine the types of electric vehicles available in the market to understand the variety of EV offerings, including passenger cars, commercial vehicles, and alternative vehicle types, such as electric bicycles or scooters.
5. Velocity (Comparison of Electric Range Averages): Compare the average electric range among vehicles based on specific periods to assess the velocity of technological advancements in EV battery technology and vehicle efficiency.
6. Velocity (Analysis of EV Adoption Trends by Model Year): Analyze EV adoption trends over time, specifically by model year, to understand the velocity of EV market growth and the rate of introduction of new EV models.
7. Total Count on EV vehicle sale till the date.
8. Value (Exploration of Price Sensitivity): Explore price sensitivity among consumers to assess the value proposition of EVs compared to traditional internal combustion engine vehicles, examining factors such as upfront costs, total cost of ownership, and potential savings in fuel and maintenance expenses.

   By addressing these research goals, this study aims to provide a comprehensive analysis and visualization of the EV vehicle population, offering valuable insights into the current landscape, trends, and potential future trajectories of the EV market.

5 Results Summary

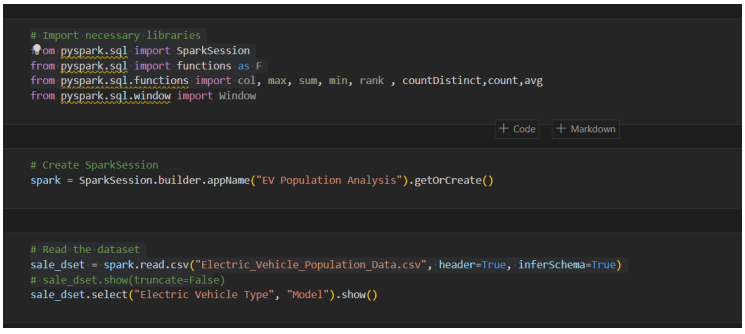
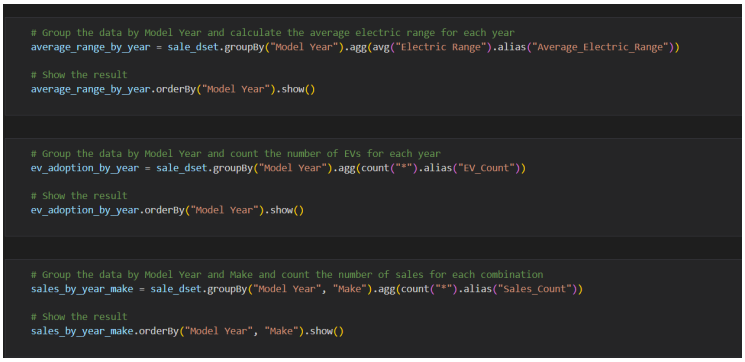
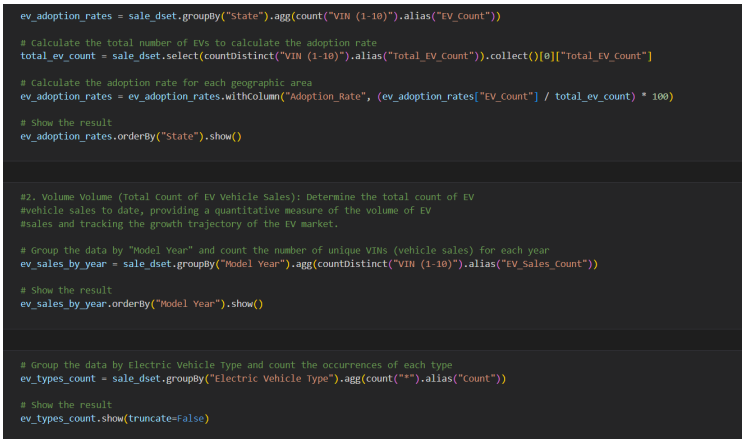


Fig. 2. Source Code



```
# Group the data by Model Year and calculate the average electric range for each year
average_range_by_year = sale_dset.groupBy("Model Year").agg(avg("Electric Range").alias("Average_Electric_Range"))

# Show the result
average_range_by_year.orderBy("Model Year").show()


# Group the data by Model Year and count the number of EVs for each year
ev_adoption_by_year = sale_dset.groupBy("Model Year").agg(count("*").alias("EV_Count"))

# Show the result
ev_adoption_by_year.orderBy("Model Year").show()


# Group the data by Model Year and Make and count the number of sales for each combination
sales_by_year_make = sale_dset.groupBy("Model Year", "Make").agg(count("*").alias("Sales_Count"))

# Show the result
sales_by_year_make.orderBy("Model Year", "Make").show()
```

Fig. 3.



```
ev_adoption_rates = sale_dset.groupBy("State").agg(count("VIN (1-10)").alias("EV_Count"))

# Calculate the total number of EVs to calculate the adoption rate
total_ev_count = sale_dset.select(countDistinct("VIN (1-10)").alias("Total_EV_Count")).collect()[0]["Total_EV_Count"]

# Calculate the adoption rate for each geographic area
ev_adoption_rates = ev_adoption_rates.withColumn("Adoption_Rate", (ev_adoption_rates["EV_Count"] / total_ev_count) * 100)

# Show the result
ev_adoption_rates.orderBy("State").show()


#2. Volume Volume (Total Count of EV Vehicle Sales): Determine the total count of EV
#vehicle sales to date, providing a quantitative measure of the volume of EV
#sales and tracking the growth trajectory of the EV market.

# Group the data by "Model Year" and count the number of unique VINs (vehicle sales) for each year
ev_sales_by_year = sale_dset.groupBy("Model Year").agg(countDistinct("VIN (1-10)").alias("EV_Sales_Count"))

# Show the result
ev_sales_by_year.orderBy("Model Year").show()


# Group the data by Electric Vehicle Type and count the occurrences of each type
ev_types_count = sale_dset.groupBy("Electric Vehicle Type").agg(count("*").alias("Count"))

# Show the result
ev_types_count.show(truncate=False)
```

Fig. 4. Source code

By addressing these research goals, this study aims to offer valuable insights into the EV Adoption Rates, Sales Count, Vehicle Types, Range Comparison, Adoption Trends, Price Sensitivity: Comprehensive analysis of electric vehicle landscape.

This PySpark code analyzes an Electric Vehicle (EV) population dataset by examining adoption rates, sales trends, and vehicle characteristics. It calculates EV adoption rates by state, total EV sales by year, counts of different EV types, average electric range by year, EV adoption counts by year, and sales counts by year and make. Utilizing PySpark's DataFrame API, it efficiently processes the dataset to provide valuable insights into the growth and dynamics of the EV market.

6 Conclusion

In summary, this PySpark project, which has been integrated with Tableau for visualization, provides a detailed analysis of an Electric Vehicle (EV) population dataset. The project sheds light on the growth and development of the EV market by looking at EV adoption rates, sales trends, and vehicle characteristics. The analysis utilizes PySpark's DataFrame API to efficiently process large amounts of data, allowing stakeholders to make informed decisions about EV adoption, infrastructure investment, and market strategy. Overall, this project emphasizes the importance of advanced analytics tools in extracting actionable insights from big data, which drives progress in the electric vehicle industry.

References

1.Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., Stoica, I. (2010). Spark: Cluster computing with working sets.
2.Electric Vehicle (EV) Data: California Air Resources Board. (n.d.). Electric Vehicle Population Data. Retrieved from
For future reference https://ww2.arb.ca.gov/our-work/programs/electric-vehicle-population-data
3."Electric Vehicle" by Union of Concerned Scientists Website:https://www.ucsusa.org/resources/electric-vehicles-evs
4."Electric Vehicles: Technology, Market, and Policy Issues" by Congressional Research Service Website:https://fas.org/sgp/crs/misc/R42502.pdf
5."Electric Vehicle Market Outlook" by International Energy Agency (IEA) Website: https://www.iea.org/reports/electric-vehicle-market-report-2020
6."Electric Vehicle Battery Technology" by Battery University Website: https://batteryuniversity.com/learn/article/electric_vehicle_ev
7. My Github URL : https://github.com/TrinadhM-dev/EV_PopulationDF.git