Analysis of impact of socio-economic indicators on Micro Transit Occupancy

Name: Samir Amitkumar Gupta

Team 17 – Project Report

Subject: CS5266 - Topics in Big Data

I) Introduction

The goal of this project is to analyze big data primarily Micro transit e-scooter data and gain insights from it. The hypothesis for this project is that socio-economic factors have a direct proportionality to the e-scooter usage in Davidson County. The motivation behind this project is that there are socio economic factors that govern the usage of scooters area wise. Socio economic factors can include median household income, and other indicators of income as well. The data for socio economic factors is readily available online using census website for United states of America. Further, the goal of this project is also to demonstrate the potential ability of big data technologies such as Spark. In this project I process 3.5 Giga bytes of trip data using Spark on a machine with 16 GB RAM and Intel i7 – 11th Gen (4 Core processor). In a real-world situation not all the 16 GB RAM is available for processing and loading data into RAM (which was the case in this project).

I had initially attempted to load all the data into the RAM using Pandas DataFrame, however that was unsuccessful as it led to out of memory errors and system crashes. That is where Apache Spark came to the rescue, I was able to successfully partition the data to process on my 4-core machine and process and perform various transformations on the data. I made use of various powerful tools in Spark such as User Defined Functions (UDFs) and Structured Query Language (SQL) tools available inside of Spark.

In this project I also replicated my code using cloud technologies on Amazon AWS cloud service, namely, Amazon Athena, EMR cluster and S3 buckets. I discuss in the later section's the issues faced by me when using these technologies and use cases of these technologies in the later sections.

From this project I was successfully able to analyze big data and create various visual plots that can help guide further research and optimization of the placement and utilization of e-scooters in the city of Nashville. Effectively the knowledge and insights gained from the visualizations of this project can further utilized in various other cities and it can be identified whether a similar trend is happening. This project can also help guide optimizing the placement of charging stations by placing charging stations in areas for riders to end their trips in those areas.

II) Dataset Used and Hardware Used:

I made use of three main datasets for this project:

a) E-scooter availability data (from Bird given to me by Dr. Dubey).
b) E-scooter trips data (from Bird given to me by Dr. Dubey).
c) Socio Economic data (for USA).

Datasets (a) and (b) were for Davidson County, Nashville and their size was 400 Mega Bytes and 3.5 Giga Bytes respectively. The socio-economic data consisted of data for the entire United states of America. I filtered out the data and used only the data for Davidson County as that was the area of interest. The size of this filtered data was 5 Mega Bytes.

The hardware specification for this project is below:

a) CPU – Intel i7 11th Gen (4 core laptop processor)
b) GPU – Intel Iris Xe
c) RAM – 16 GB
d) Storage – 1 TB SSD storage.
e) Cloud Technologies – Amazon AWS Learner Lab

III) Technology Used:

To complete this project, I made use of Python as my coding language, the python version used was 3.9.13. I made use of python as it provides an easy coding interface to work with over Spark, Pandas and Geopandas, I will explain the advantages of these technologies later in this section. Further, python has useful libraries for creating visual plots using libraries such as plotly, matplotlib and seaborn. Below is an introduction of all the technologies used:

i) Spark

Apache Spark developed at UC Berkeley AMPLab in 2009 [1] is a powerful distributed computing framework [1]. This open-source technology makes it possible to quickly analyze and handle enormous volumes of data [2]. Spark is widely used in the industry today, by various companies for managing and handling big data, it is heavily used in sectors that deal with a lot of time series and big data such as Finance, Healthcare, and retail [3]. Through a language-integrated API akin to DryadLINQ in Scala, a statically typed functional programming language for the Java VM, Spark offers the RDD abstraction [1]. Spark has the ability of lazy computation, i.e., using two main concepts transformations and actions, it only runs the transformations on the data whenever an action is called on it. Until then spark goes on to build the RDD (Resilient Distributed Dataset) using DAGs (Directed Acyclic Graphs) [1]. There are various applications and benefits of using Spark such as Real-time data processing – Spark can be heavily used for processing of data from IoT device sensors [3]; Large Scale Data Analysis - Spark can handle data sets of several terabytes or petabytes, making it an excellent choice for big data applications[3]; Machine Learning – Spark has a machine learning library called MLlib that can be used to run various machine learning algorithms for classification, regression and clustering; Stream Processing – this is a very powerful tool that can be used for real time stream processing of data from various input sources such as PubSubClient, Kafka Messaging System, Databases and even IoT sensors [4].

Benefits of using Spark include [1]; High Performance - Spark is built to operate quickly and effectively, it is a great option for processing and analyzing massive amounts of data since it can carry out computations up to 100 times quicker than Hadoop; Fault tolerance: Spark is built to gracefully manage errors. Data processing and analysis can continue uninterrupted since it can recover from node failures or network partitions; Cost-effectiveness: Spark is a free to use open-source framework. Commodity hardware, which is considerably less expensive than proprietary gear, can be used to install it and Finally; Simple API: Spark's straightforward API makes it simple for programmers to create sophisticated data processing and analysis applications. Additionally, it supports a variety of programming languages, such as Python, Java, and Scala. For this project Spark played a crucial role wherein it was used to read and process the 3.5 GB e-scooter dataset.

ii) Pandas

Pandas is a robust toolkit for Python programming that allows for data manipulation and analysis [6]. It offers data structures and operations for successfully managing and adjusting huge and intricate data collections [7]. Pandas can read data of any types that includes files of type JSON, parquet, pickle, text, etc. The main features of Pandas include; Data structures – includes two main types series and dataframe (looks similar to tables in SQL) [7]; Data Manipulation – pandas includes various functions for sorting, filtering, merging and reshaping, it also has the ability to make use of SQL query over the dataframes; Time series Analysis and integration with other libraries – pandas can be easily integrated with other libraries such as numpy and pyspark (can make udf's that can be used with big data).

iii) Geopandas

GeoPandas adds support for geographical data to the well-known data science library pandas [8]. It provides pandas the ability to read geometry data that is present in geospatial data. This data can be viewed inside a python notebook easily using this library.

iv) Plotly

An interactive, open-source plotting toolkit for Python, plotly provides over 40 different chart types for a variety of statistical, financial, geographic, scientific, and three-dimensional use-cases [9]. It is built on top of the JavaScript library (plotly.js) and can be used to create various plots and saving them as a HTML useful in web applications, PNG, or various other formats. Most of the plots for this project were created using plotly, the integration of plotly with mapbox (explained later) was key for creating visualizations in this project.

v) Mapbox

The robust data visualization tool Plotly Mapbox enables users to build dynamic maps with numerous layers of data [10]. It is based on the Mapbox mapping platform and offers several tools for designing beautiful and useful maps [11].

vi) Cloud Technologies Used

In this project, I made use of Amazon Web Services cloud service available using AWS Learner lab that provides 100$ free credit for students in the Topics in Big Data class for the Spring 2023 semester. Amazon AWS is a very powerful cloud service offering various technologies for dealing with Big Data namely EC2 instances, wherein one can create a virtual machine with say 100 GB RAM and can be used to read and process Big Data. However, for this project I made use of three very useful technologies offered by AWS:

A) Athena: Using conventional SQL, it is simple to evaluate data directly in Amazon Simple Storage tool using Amazon Athena, an interactive query tool [12]. Athena can be used to use your data stored in Amazon S3 and start using conventional SQL to conduct ad-hoc queries and obtain answers in seconds with a few clicks in the AWS Management Console [12]. This is a very powerful tool that provides query results on Big Data of the size of petabytes in time ranging between seconds and minutes. This tool was used in this project as to check the potential of this tool on big data of size 3GB.

B) EMR Cluster:

The cluster is the main part of Amazon EMR. Amazon Elastic Compute Cloud (Amazon EC2) instances are grouped together to form clusters [13]. A node is any instance inside a cluster [13]. The node type describes the function that each node plays inside the cluster. Each type of node is given a role in a distributed application like Apache Hadoop by Amazon EMR, which also installs various software components on each node type [13].

C) S3 Bucket:

An object storage service called Amazon Simple Storage Service (Amazon S3) provides performance, security, and scalability that are unmatched in the market [14]. For a variety of use cases, including data lakes, websites, mobile applications, backup and restore, archives, enterprise applications, IoT devices, and big data analytics, customers of all sizes and sectors can use Amazon S3 to store and protect any amount of data [14]. In this project all the data was stored in S3 bucket and data was easily accessed by EMR clusters and Amazon Athena with the help of S3 Buckets.

IV) **Methodology and Visualization**

This section highlights the methodology used for the project and the visualizations created for the analysis of E-scooter and socio-economic indicator data.

The data was first preprocessed/cleaned, this included fixing the format of various columns in the data; fixing the data type of various columns, for example, certain columns had data type as a list and had only a single element inside the list such data types were fixed and replaced with a string or integer depending on the data type of the element in the list; further the data was of JSON format which stored everything as a key value pair certain key-value pairs were broken and only the value part was retained (this was not done for geometry column); Geometry column was fixed and converted to shapely geospatial data format – this was done to ensure that the data was readable in GeoPanda dataframes. Finally, only those columns were selected from the dataset that were required for the final visualization and processing in this project. Processing of socio-economic data only included selecting columns that were required i.e., GEOID, Locations, median income, and geometry columns. Finally, all the duplicate records were removed from all the datasets

Once the preprocessing was complete, visualizations were created on the GeoPanda dataframes using plotly and matplotlib. The starting points of each trip was also determined for analyzing the occupancy per area for each trip, the starting point was then joined with the socio-economic dataset was determining the area under which the trips started, this was achieved with the help of a spatial join (sjoin) available in geopanda's library. Below I explain each visualization created as part of this project.

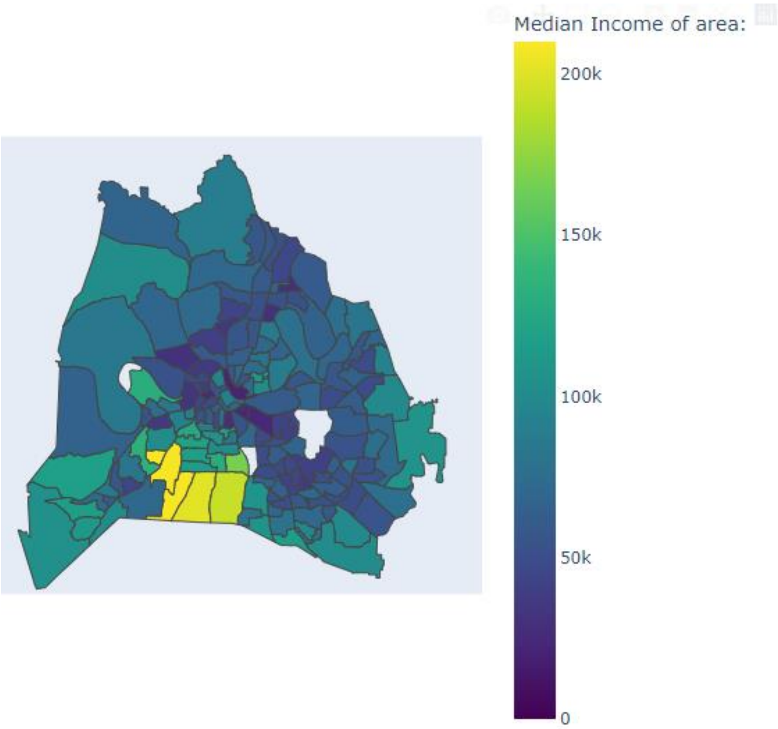


Fig (1): Socio-eco indicator

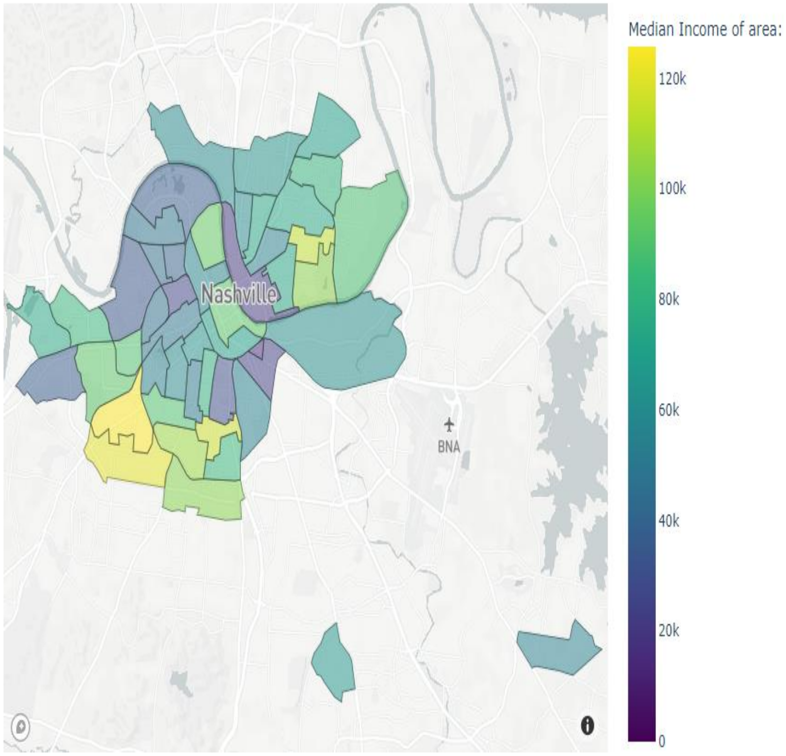


Fig (2): Socio-eco indicator only trips

The figures (1) and (2) are visualizations for socio-economic indicators for Nashville Davidson county. In both these graphs the legend indicates the median of each area ranging from 0 dollars to 120,000 dollars. Figure (2) is a subset of fig (1) it only contains those area for which trip data was available. This graph was merged with trip start data points for further visualization.

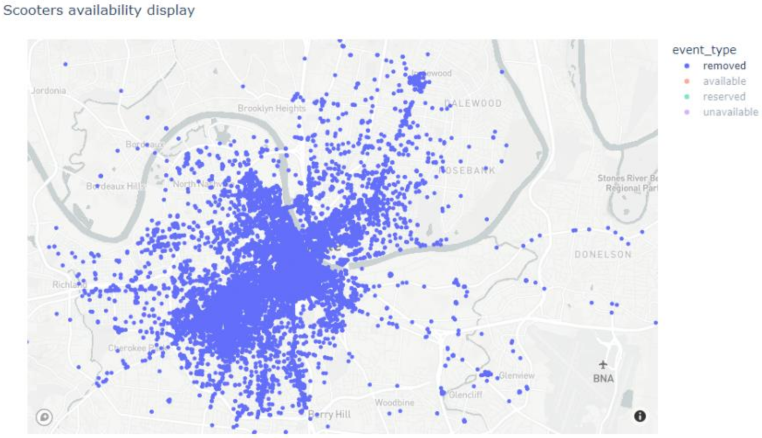


Fig (3): Scooter's removed visualization

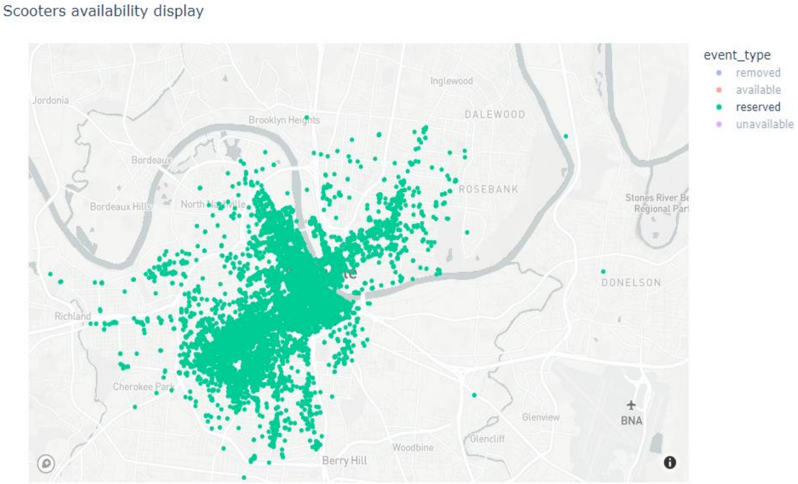
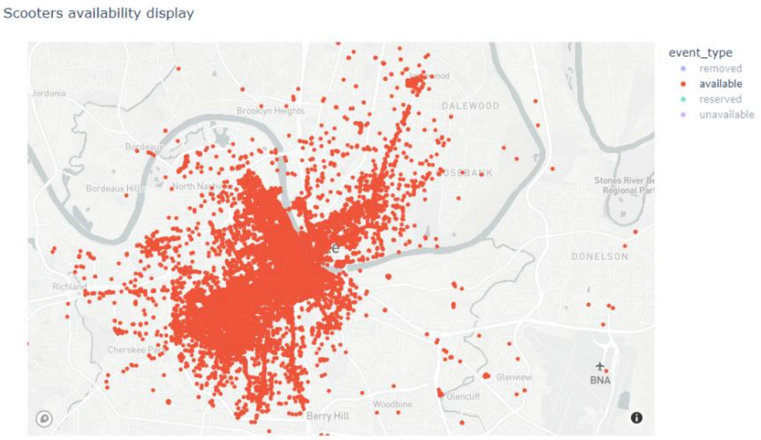


Fig (4): Scooter's reserved visualization



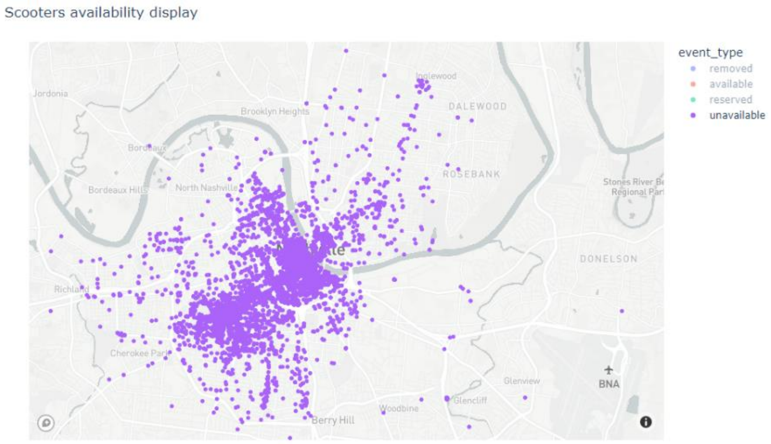


Fig (5): Scooter's available visualization

Fig (6): Scooter's unavailable visualization

Figures (3), (4), (5) and (6) are used to indicate the scooter availability in Davidson County. This includes scooters that were removed (Fig 3) these can include scooters that were removed due to some physical issue with the scooter such as a non-working battery, physical problem with the scooter such as a broken handle or problem with the braking system or even some technical problem such as software glitches. Fig 4 depicts scooters that are reserved at any time of the day. Fig 5 shows all the scooters that are available throughout Davidson County across the entire life time of the scooters. Fig 6 indicates scooters that are not available at any given time. This can happen due to issues such as no battery in scooter or a software update required for the scooter, etc.

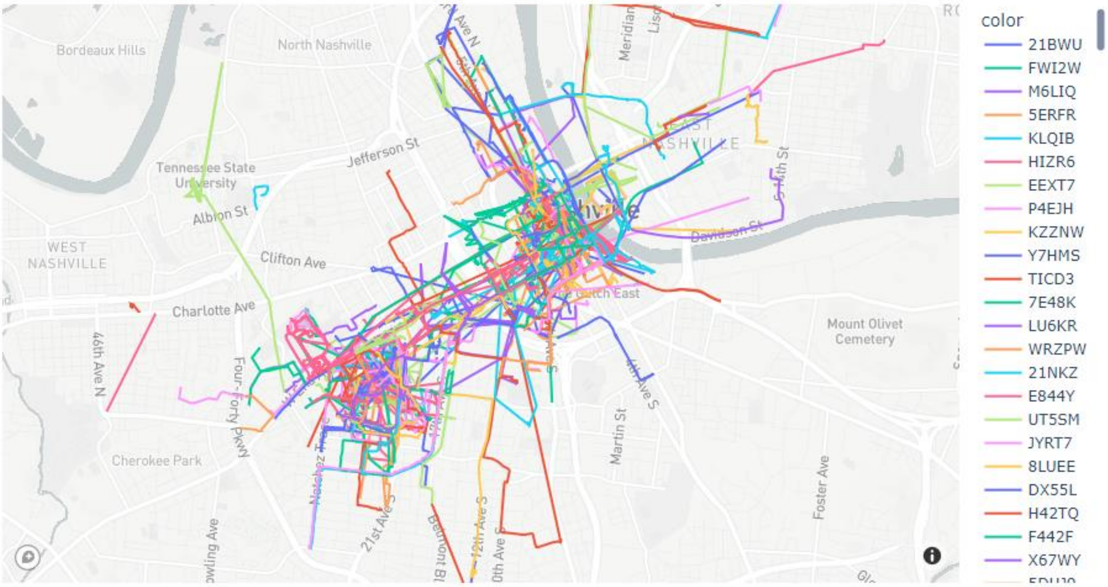


Fig (7): 500 E-scooter Trips in Davidson County

Figure (7) shows 500 random trips taken by e-scooter users in Davidson County. For visibility not all the 97000 trips were plotted on the graph. This graph was plotted using plotly. The legend in the graph corresponds to the ID associated with each e-scooter. There can be like with the same color as a single scooter can be used by multiple users at different times. The starting position of these can change as well as the scooters are manually picked up and charged by the scooter provider.

V) **Results Obtained**

The results obtained in this project were coherent with the hypothesis for the project i.e. higher median income indicated more e-scooter usage. This statement is supported with the help of Figures (8) and (9) below.

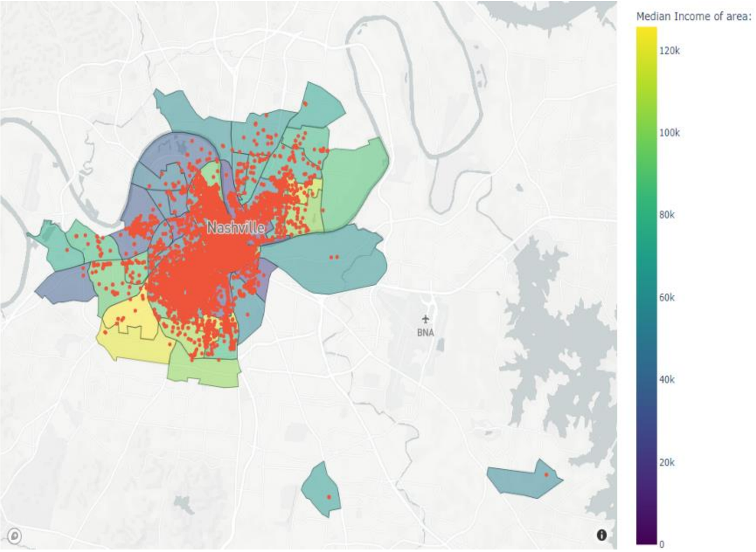


Fig (8): Socio-eco and trips merged

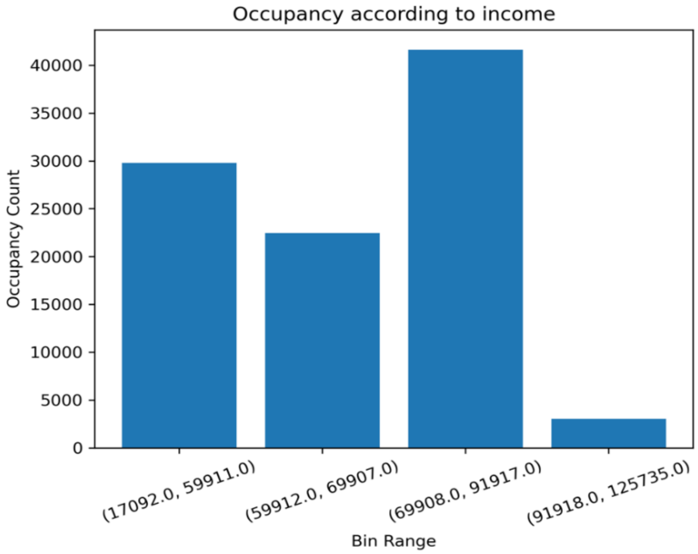


Fig (9): Occupancy of e-scooter

In figure (9) the socio-economic median income was divided into 4 bins:

1) (17092.0, 59911.0) – Low
2) (59912.0, 69907.0) – Medium
3) (69908.0, 91917.0) – High
4) (91918.0, 125735.0) – Very High

The trend for occupancy can be easily understood by referring figures (2) and (8). The occupancy for bin range 3 is the highest indicating high occupancy where socio economic factors are high and from bins 0 and 1 low socio-economic factors co-relates to low e-scooter usage.

The important question is “why does bin 4 has low e-scooter usage?” as it completely breaks the hypothesis made for this project. The reason for that can be understood from figure 2, as we can see that bin 4 mainly consists of area that are away from the city center, areas where people tend to use cars are their primary means of travel as they primarily travel long distances and fig (5) supports this hypothesis wherein there is low availability of e-scooters in areas away from the city center.

VI) Use Cases and Future Work

The work done in this project can help drive further research in the optimization of e-scooter placement and charging. Future work also includes how weather, office hours and holidays play a role in e-scooter usage. The above-mentioned data can be visualized further to determine further insights into usage of e-scooters.

This work can also be further expanded to understand usage in other cities around USA as well. Another potential research area is cost optimization of e-scooters, this problem can make use of weather data and an online generative model can be created with the help of Reinforcement learning to optimize the cost charged to users.

This project can be further expanded by analyzing the end locations near the city center for creating charging stations for these e-scooters, where users can end trips at charging stations close by.

VII) Lesson Learnt

The lessons learnt by this project include below:

1) Big Data is difficult to work with when you have major memory constraints.
2) Lot of times I ran into issues with memory – I ran into various issues when using pandas for reading data and gave me motivation for using other technologies for processing and reading of data.
3) Spark comes to the rescue – Spark is the best solution for processing big data for systems that do not have enough memory to read big data. Spark is extremely fast and easy to use.
4) Cloud is also useful but not always – Cloud is extremely powerful; however, it can be complicated to work with sometimes, wherein, I was not able to get EMR cluster to run the Spark code written by me. Athena is extremely powerful as well, however setting up the table and the entire setup can be very time-consuming and expensive too.

VIII) References

1) Zaharia, M., et al. (2012). Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (NSDI 12).
2) Apache Spark https://spark.apache.org/.
3) Kumar, R. and Kant, R. (2017). Big Data Analytics with Spark. Cham: Springer.
4) Farooq, U. and Sultana, S. (2018). Stream Processing with Apache Spark. Birmingham: Packt Publishing.
5) McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.
6) Reback, J., et al. (2020). Pandas Development Team. pandas-dev/pandas: Pandas, version 1.1.4.
7) VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.
8) https://geopandas.org/en/stable/getting_started/introduction.html
9) https://plotly.com/python/getting-started/
10) https://plotly.com/python/maps-mapbox/
11) https://www.mapbox.com/

12) https://docs.aws.amazon.com/athena/latest/ug/what-is.html
13) https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-overview.html
14) https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html