A project report on

DIVVY BIKE-SHARE ANALYSIS FOR TARGETED CUSTOMERS MARKETING

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

By

TAMOJIT ROY (19BCE1156)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

DIVVY BIKE-SHARE ANALYSIS FOR TARGETED CUSTOMERS MARKETING

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

By

TAMOJIT ROY (19BCE1156)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I here by declare that the thesis entitled "DIVVY BIKE-SHARE ANALYSIS FOR TARGETED CUSTOMERS MARKETING" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai, is a record of bonafide work carried out by me under the supervision of Dr. A. Sheik Abdullah.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: Signature of the Candidate

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"Divvy Bike-Share Analysis for Targeted Customers Marketing"** is prepared and submitted by **Tamojit Roy** (**19BCE1156**) to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** programme is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. A. Sheik Abdullah

Date:

Signature of the Examiner 1

Name:

Date:

Signature of the Examiner 2

Name:

Date:

Approved by the Head of Department
B. Tech. CSE

Name: Dr. Nithyanandam P

Date: 24 – 04 – 2023

(Seal of SCOPE)

ABSTRACT

Although the age calls for motor-vehicles as the major shareholders of the commute industries, cyclists still continue to have impressive shares in some of the developing heavyweights and developed nations around the world. Some of the countries like Sweden, Denmark, Germany, UK, Japan and even China have a wonderful landscape for cycling. Although some of them are casual riders, quite a many are annual members for the major cycle manufacturing companies all over the world. Data analytics has a marvellous job to play in analyzing and boosting the sales of bike share plans of any good company specializing in providing BSS. It has an upper hand when it comes to implementing market plans to target the set of customers who are the most vulnerable by suggesting them various specialized schemes and membership benefits. Hence, it is no doubt one of the most potent tools helpful in boosting the sales of a product. Hence, it can play a pivotal role in increasing the annual membership of cycles of any company for good profits. This research work intensely focusses on analyzing all the major aspects and most of the if not all of the attributes of bike-share sales of a prominent bike-share company in Chicago named Divvy. This research work mainly revolves around understanding how subscribers and customers of Divvy bike-share service use bikes differently. The comparison along with other tasks have been used to design marketing strategies aimed at converting customers of the company to the subscribers of its services.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. A. Sheik Abdullah, Assistant Professor Senior Grade 2, SCOPE, Vellore Institute of Technology, Chennai, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Data Analytics, Soft Computing, Machine Learning, Swarm Intelligence, Clinical Informatics.

It is with gratitude that I would like to extend thanks to our honorable Chancellor, Dr. G. Viswanathan, Vice Presidents, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan and Mr. G V Selvam, Assistant Vice-President, Ms. Kadhambari S. Viswanathan, Vice-Chancellor, Dr. Rambabu Kodali, Pro-Vice Chancellor, Dr. V. S. Kanchana Bhaaskaran and Additional Registrar, Dr. P.K.Manoharan for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dean, Dr. Ganesan R, Associate Dean Academics, Dr. Parvathi R and Associate Dean Research, Dr. Geetha S, SCOPE, Vellore Institute of Technology, Chennai, for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Nithyanandam P, Head of the Department, Project Coordinators, Dr. Abdul Quadir Md, Dr. Priyadarshini R and Dr. Padmavathy T V, B. Tech. Computer Science and Engineering, SCOPE, Vellore Institute of Technology, Chennai, for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai, who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date:

Tamojit Roy

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BSS	Bike Share Service
SWOT	Strength Weakness Opportunity Threat
IFEM	Internal Factor Evaluation Matrix
EFEM	External Factor Evaluation Matrix
NMT	Non Motorized Transport
GHG	Green House Gases
FDI	Foreign Direct Investment
FFNN	Feed Forward Neural Network
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
MLP	Multi Layer Perceptron
SVR	Support Vector Regression
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
SMAPE	Symmetric Mean Absolute Percentage Error
CNN	Convolutional Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
PSS	Product Service System
DBS	Dockless Bike Sharing
GAM	Generalized Additive Model
ARMA	Auto Regressive Moving Average
SLR	Systematic Literature Review
PRISMA	Preferred Reporting Items for Systematic reviews and Meta-Analyses
MNL	Multinomial Logit
MaaS	Mobility as a Service
ICT	Information and Communication Technology
SCE	Stated Choice Experiment
CAGR	Compound Annual Growth Rate
MSE	Mean Square Error
ADF	Augmented Dickey Fuller
CRAN	The Comprehensive R Archive Network
GBDT	Gradient-Boosted Decision Tree
NBEATS	Neural Basis Expansion Analysis for Time Series
AIC	Akaike's Information Criterion
BIC	Bayesian Information Criterion
FPE	Akaike's Final Prediction Error
HQIC	Hannan-Quinn Information Criterion

Chapter 1

Introduction

1.1 INTRODUCTION

In the world, the present dynamic state that it is in, a significant lot of people beguiled by their work ethics and cultures need to travel a lot. This poses a serious challenge especially for those who rely on commute services to garner opportunities and those who seek major success in their careers. The competition becomes quite intense in regions with dense but competitive population. In such a case, the need for a system to alleviate these kind of problems arises.

1.2 BIKESHARE SIGNIFICANCE

Congested spaces pose a murky hindrance to those who are struggling both socially as well as economically, as they have limited options when it comes to choosing commute services. In many cases, they are not aware of various bike-share services which can alleviate their suffering.

1.3 COMMERCIAL SHORTCOMINGS

The bike-share companies, on the other hand are not familiar with the needs of such sections and hence fail to fabricate plans to come up with customer-centric policies and subscriptions for boosting sales and for providing help as well. They fail to realise the potential this industry actually has in reality and hence they fail to plan accordingly and take serious measures to provide for the industry and build solid frameworks and structures which can perform efficiently.

1.4 PROBLEM STATEMENT

Understanding how subscribers of Divvy bike-sharing trips and the bike-sharing service availing only-customers use the bikes differently for commute purposes, recreational purposes, schooling, marketing, etc. and analyzing the multifarious shadow factors which influence both the user categories in opting for the type of service which they show their inclination towards can be a really challenging task. Figuring this comparison along with other tasks which may later be used to design marketing strategies aimed at converting the customers of the company called Divvy to the dedicated major league consumers or patrons of the bike-sharing service is another problem which lacks an efficient way to be addressed.

1.5 WHAT CAN BE DONE

The first and foremost imperative move is to develop a research work which mainly revolves around understanding how steadfast patrons or subscribers and regular customers use Divvy bikes differently. The comparison along with other tasks can be made to good and productive use of for scheming or formulating various kinds of advanced marketing strategies and plans targeted towards the regular customers in order to flip their soul and personality to that of a faithful and dedicated subscriber of the company.

1.6 ABOUT RESEARCH

This research work is focussed exactly to address the same problem by taking the instance of Divvy's bike share system. Divvy is a well-known and ubiquitous name and is well reputed to provide bike sharing facilities across Chicago and Evanston. Understanding how subscribers of Divvy bike-sharing trips and the bike-sharing service availing only-customers use the bikes differently for commute purposes, recreational purposes, schooling, marketing, etc. and analyzing the multifarious shadow factors which influence both the user categories in opting for the type of service which they show their inclination towards can prove to be of immense help. This comparison along with other tasks has been used to design marketing strategies aimed towards converting the customers of Divvy BS trips to the major league faithful subscribers of the BSS.

1.7 SCOPE OF THE PROJECT

Although the age calls for motor-vehicles as the major shareholders of the commute industries, cyclists still continue to have impressive shares in some of the developing heavyweights and developed nations around the world. Some of the countries like Sweden, Denmark, Germany, UK, Japan and even China have a wonderful landscape for cycling. Although some of them are casual riders, quite a many are annual members for the major cycle manufacturing companies all over the world. As data analytics is ubiquitious nowadays, harnessing its power to help improve the conversion of casual customers to dedicated subscribers of any bike-share company can be of paramount importance to the company in boosting its profits.

Chapter 2

Literature Review

S.No	Paper Title	Summary	Algorithms Used	Pros/Cons
[1]	Bicycle industry as a post-pandemic green recovery driver in an emerging economy: a SWOT analysis	This paper deals with analyzing and examining the various factors, which portray the bicycle-industry as a potent industry which can lead to the green recovery and sustainable development of economy and environment in Bangladesh. The authors performed a SWOT analysis after collecting and analyzing information with regards to the bicycle industry in the south-Asian developing economy from many research publications, and by interviewing industry experts, government officials and university professors and finally a joint discussion by all the experts. Based on the findings in which both internal factors (strengths and weaknesses) as well as external factors (opportunities and threats) were taken into consideration, they came up with an internal factor evaluation matrix (IFEM) to find out that the bicycling industry in Bangladesh has enormous potential, given that some reforms be conducted upon the same with a view towards checking pollution arising solely because of the naughty element addressed by the name of carbon in the post-pandemic settings. Suggested potential strategies to invigorate the acquisition of	SWOT, IFEM, EFEM.	Pros: 1. Can be used for making strategic planning decisions. Cons: 1. Data collected for analysis can never be sufficient and reliable.

		bikes as a sexy option for transportation by encouraging many many good options such as by commencing manufacturing businesses in local stretches of land, dedicated infrastructure, reducing import duties, attracting FDIs, eliminating gender differences, etc., were also mentioned.		
[2]	Bike sharing usage prediction with deep learning: a survey	The analysis of the pattern of bike-sharing in a region plays the most significant role in predicting the bike usage in any particular region. The authors in this paper, have provided quality review with many invaluable approaches to predict the BS usage pattern with DL. Following a set of procedures of the following modules: data aggregation, defining the 3 formats of data, addressing the 3 types of prediction, quantifying the prediction error using different evaluation metrics, and finally some prediction challenges (complex spatial dependencies and complex temporal dependencies) and prediction models like FFNN, LSTM, RNN, GNN, GRU, MLP, SVR, etc., were illustrated with a different section for each type of model was mentioned for both the BSSs. Finally, application scenarios within the bike-sharing systems and beyond were brought up followed by other vital things.	Fuzzy c-means clustering, distance-based clustering, RMSE, MAE, MAPE, SMAPE, RNN, CNN, GCN, GAT.	Pros: 1. Good prediction accuracy when unexpected events are kept out of the bay. Cons: 1. Unexpected variability of external factors.
[3]	An Analysis of a Bike-Sharing System from a Business Model	A vision of developing a sustainable bike-sharing system as viewed a PSS (product-service system) was	PSS.	Pros: 1. PSS solution offered.

	Perspective	expressed by the authors in this research paper. Although, it was initially developed considering a single focal-company, restricted to a particular region (southern region of developing country, Brazil) only, the authors emphasized the significance of their research-work claiming that it was the first one to be done keeping in mind the scenario of a developing country. The design of the system-model organized in 4 stages was introduced. Although, the strategy doesn't hold the providers of the service as the manufacturers themselves, the model which is utilized for performing business had enormously big amount of things to do with shared mobility. Those who in reality developed the business were taken to an interview room with face-to-face facilitation to grow a protocol of research. The authors expressed that the PSS business model analyzed by them could prove to be colossally enriching micro-mobility contributions.		Cons: 1. No significance of validation flowing from external sources as the focal company only is kept in spotlight in a singleton study.
[4]	Bike Share Usage and the Built Environment: A Review	This paper entirely deals with understanding, analyzing and illustrating the multiple modes and forms of relationships between build environment (i.e., land-use, transportation system and urban design) and bike-share usage. Quite a many variances between the build environment and bike-usage were stated and described with some outliers in notable cases.	Self-devised algorithm for literature search and selection process.	Pros: 1. Among the very few selected studies which were empirical, there was inconsistenc y in the way BS usages were

		Variance in relationship in the build environment across different mobility patterns, docked and dockless bike-share patterns, w.r.t. trip purpose, between arrival and departure patterns, based upon the day of week, etc. and the bike-share usage were elaborated. The paper concluded with a brief summary of the major findings of the authors and them encouraging the recommendations for the future research works.		influenced by some rarest of the factors of built environment's effects. Cons: 1. Requires a more comprehensive approach for better insights.
[5]	Examining factors associated with bike-and-ride (BnR) activities around metro stations in large-scale dockless bikesharing systems	The study attempted to examine the associations of BnR (bike and ride) activities with metro area w.r.t. DBS systems, in the city of Shanghai, China. The study signalled that BnR behaviors were affected by features like station features, roadway designs, transportation facilities, etc. Mainly four metrics were employed in the entire study to understand the behaviors of bike-and-ride activities according to the different point of views of diverse participators viz. local govt., DBS users, etc. There were various metrics in usage for the assessment of BnR performance. The GAM was utilized to build statistical inference. Several statistical issues were addressed while modelling DBS usage. The spatial distribution of the 4 metrics suggested that the city center was flooded with shared bikes whereas there was a drought of shared bikes in the	Negative binomial regression, multilevel mixed model, ARMA, GAMs.	Pros: 1. A very useful all inclusive as well as detailed statistical framework was provided by the study which could prove to be very helpful in evaluating bike-and-ride performances in solid systems involving DBSmetro, relocation of support shared bike, etc. to ultimately uplift the attractiveness of utilizing DBS in realizing its

		suburb. Based on other things, various other conclusions were drawn for comprehensive analysis.		role as a feeder transit mode. Cons: 1. Correlational variables are made use of which can't prove in any reality, the causality. 2. In the complete void of valuable information concealed in the individual, at an aggregate plain, there was made-sense-of operation of all the analysis made. 3. Data represents behavior of only DBS users.
[6]	Increasing Bike-Sharing Users' Willingness to Pay — A Study of China Based on Perceived Value Theory and Structural Equation Model	In this research paper, the authors sought to investigate the correlation of the various factors of the perceived value upon the users enthusiasm to pay for BSSs in the first-tier and second-tier cities of China. A structural analysis was also conducted to validate the findings and visualize the significance of the different factors as variables. The research paper put light on the	Scale Design on PV, PU, PE, PEU, PC, PR, IPC, WOM, PT, EP, WTP; Reliability Analysis; Validity Analysis; Confirmatory Factor	Pros: 1. Assess bike-sharing users consumptive decisions. 2. Assess users' willingness to pay. Cons: 1. Model could have

		2 variants of factors namely the direct and the indirect ones that can manipulate the user using the BSSs in terms of their enthusiasm to pay. Based upon the findings, the authors concluded that, perceived usefullness and other criteria have a great impact in the positive sense in the domains of perceived trust & value, protection of the environment, etc. impact on perceived value; the users' word-of-mouth and perceived entertainment have no significance; and finally impact in the negative sense was seen on perceived value caused by perceived risk as well as cost.	Analysis and Correction; Model Building.	been more accurate and there is scope of improvement .
[7]	Understanding the intention to use bike-sharing system: A case study in Xi'an, China	This paper investigates the various factors which make the bike-sharing services to be retained by the users, and not just be opted by them in the first place. For data collection purposes, questionnaires were collected through both online and offline survey. The activity of collection of sufficient number of questionnaires was performed. The authors introduced the related concepts of participation in purchasing decisions, customer engagement, etc., and uses a structural equation model to identify the interaction and influence mechanisms among the 3 variables and usage intent. It provides management and marketing strategies for bike-sharing companies.	SEM, Reliability & Validity, Direct effect test, Intermediary effect test.	Pros: 1. Showed that all influencing factors were significantly positively associated with usage intention. Cons: 1. Limited sample size. 2. Some factors like residents' travel characteristics, weather, etc. not considered.
[8]	Bicycle Industry in India and its Challenges – A	Through the paper, the authors tried to analyze and discuss the current bicycle market scenario	Data Collection, SWOT	Pros: 1. Several enhancement

	Case Study	in India and where the developing country presently makes its stand in the world when it comes to manufacturing, exporting and ranking in terms of bikes' usage and procurement of raw materials. The objectives which they proposed in the research papers include, gaining knowledge about India's cycle industry, learning about the industry's development, comparative analysis between sales and production, major studies of industry growth, the bicycle industry's contribution in international economic development, research on how latest gadgets can be added in bikes, understanding CORONA's impact on the Indian cycle industry, and finally, a SWOT analysis to aggregate the facts and figures for recommendation purposes favoring the success of the Indian cycle manufacturing industry in the future. After collecting the data from various sources, viz., journal, published papers, archived newspaper articles, official bicycle industry websites, and other ventures, the authors discussed various aspects: the growth of bicycle industry in India, major competitors of India in the industry, etc., the examination spotlighting the SWOT analysis of the Indian bicycle industry; CSR activities; COVID-19 impact on the Indian bicycle industry. Based upon the	Analysis.	suggestions provided based on analysis. Cons: 1. Very theoretical study and very less or no numerical computation used.

		aforementioned, they provided their recommendations and concluded with areas for future development of the industry.		
[9]	Machine Learning Approaches to Bike-Sharing Systems: A Systematic Literature Review	The paper provides a SLR of various research papers specializing in exploring and analyzing the multifarious machine learning approaches applies to bike-sharing systems (BSS). Based on PRISMA methodology, the systematic literature survey was performed. A 4 phased flow diagram consisting of a multitude of different phases, was aimed to describe and understand the items of the different sections. The authors framed a process workflow to understand all the stages of the study, viz., keyword identification and search, repositories, bibliometric analysis, etc. The open source tool VOSviewer, was used for doing the analysis of network. The tool was indeed a very useful one as the main authors, co-authors and their interlinking relationships, keywords, etc. was effortlessly identified by it within the data set. A whole different types and numbers of graphs were created for each of the sections of interest, by the authors in order to bring out reasonable insights from the study. Based on the findings, the authors asserted that the 2 deadly issues pin-pointed by ML when talking about BSSs are clustering (classification) and prediction. 3 specific clustering algorithms are more commonly	PRISMA, keyword identification and search, bibliometric analysis, keyword occurence analysis, title and abstract text occurrence analysis.	Pros: 1. Outlined and identified the main ML techniques contribution to BSSs in urban mobility. Cons: 1. Many things comprising a whole diverse set of features for validation and improvisation of modelling strategies of the future as well as sufficient number of case studies are non-existent in the big fat research paper proposed.

		used. The authors additionally also discussed the various research and study limitations in their whole study.		
[10]	Factors motivating buying behavior of female two wheeler users in the district of Palghar	The research paper examines the various factors influencing the buying of two wheeler vehicles by the females in a distant suburb of Mumbai city known by the name of Palghar. For collecting the data, the authors used structured questionnaires (primary data) and a diverse set of other secondary data sources. For their research purposes, the authors collected data samples from a total of 150 respondents. Random sampling was used for the collection of primary data. The authors specified a set of hypothesis and performed a data analysis report, which put light upon the various aspects of the female buying behavior and more, viz., popularity of brands among female riders, usage of the mighty internet in formulating decisions which has to do with purchasing stuffs, awareness of celebrity endorsement, mode of purchasing the vehicle, etc. They concluded that necessity was the most influential factor affecting the buying decision of users who are female finding orgasm in driving 2 wheeler vehicles. Based on the calculations, as the Chi square value was significantly high, it made them to accept the alternate hypothesis which they set. Other conclusion were drawn based on a multitude of	Some data analysis techniques.	Pros: 1. Some insights provided. Cons: 1. Limited data.

		diverse other factors.		
[11]	Factors Influencing Purchase of Two Wheeler - A Study with Reference to Chennai City	Information from survey of local respondents and other sources were used for studying the factors which influences the purchase of two wheeler modes of transport in the city of Chennai. Both primary and secondary data were involved in the study by the authors. The primary data was collected from the owners of two-wheelers using questionnaires. The secondary data was sourced from many places. About 100 respondents were selected for collecting the primary data with suitable sampling techniques. Some of the objectives put forward by the authors included: to analyze the age of the respondents, to analyze the factors influencing the purchase of two-wheeler, and to know the expectation of consumers in the purchase of two-wheeler. This was followed by the analysis and implementation. The authors concluded the research paper with suggestions and conclusion.	Data collection, analysis and interpretation.	Pros: 1. Data collected directly from real-valued opinion of people. Cons: 1. Limited to only a very particular region (not wide-scale). 2. Simplistic data analysis and discussion overview provided.
[12]	Unlimited-ride bike-share pass pricing revenue management for casual riders using only public data	The authors scrutinized a lot of strategies which are integral for managing revenues for unlimited usage BS scenarios in their research paper. Citi Bike public data has been used by the authors for the overall analysis. Summarization of the basic data for understanding the behavioral patterns of the casual users and the type as well as amount of trips which the users that are casual i.e.,	Linear regression, multinomial logit model (MNL).	Pros: 1. Provides a good example for management of revenue in MaaS. 2. Pin points exciting insights on what can't and can be executed

		the customers select or take was prophesied or predicted by estimation technique for relating between the two, because public appearance of such data was non-existent. Using the sexy technique of linear regression which is very well known in the gigantic world, by utilizing it for short term daily ticket sales and occassional passenger numbers, the distribution parameters were extracted from sample mean as well as sd obtained as a result of using the sexy technique. A path choice model was built using variables resampled from the fitted distribution by the bootstrap method. The MNL model was used because it could represent aggregated market shares alongside individual vote probabilities. As a result, the revenue could be maximized and the impact on the surplus of the consumer was also quantified.		with the strength and endurance of big data availability. Cons: 1. Not many alternatives provided for pass choice model. 2. Estimated demand model not that comprehensive.
[13]	An approach to modeling bike-sharing systems based on spatial equity concept	This paper deals with providing a bunch of meaningful examples which are also illustrative and performing a series of tests for finding out the arrangement as well as the quantity of bike stations and unoccupied racks by doing sensitivity analysis to put forward a real and practical original model for modelling BSSs which stand upon the very concept of spatial equity aiming towards reducing implementation and operational cost of the existing such systems. In the research	A proposed linear mathematical model algorithm.	Pros: 1. Bestows a novel model required for the dimensioning of BSS which is termed public. Cons: 1. Functionality of tested model limited to a network of

		paper proposed by the authors, determination of arrangement as well as quantity of bike stations and unoccupied racks for bike parking purposes available at each station was done after developing a LPM to set up and operate the BSS for each level of defined service, terms and conditions cost was minimized. The proposed linear mathematics problem minimized the structure as well as the implementation charges of a brand new BSS that reflect the concept of a level of service that is fair and balanced by utilizing a bunch of constraints and that is made accessible to all the users of the system.		smaller dimensions.
[14]	Optimizing Bike Sharing Systems: Dynamic Prediction Using Machine Learning and Statistical Techniques and Rebalancing	Cities all around the world are increasingly getting concerned about the detrimental results of ever escalating quantities of automobiles on their roads, viz., greater pollution, emission rates, congestion, etc., which has led many of then to take active steps in order to prevent such things. One of the most effective and efficient alternative and life-saver which has come out from the horizon in this regards is the usage of BSSs. Many cities have also been seen to adopt BSSs to tackle such growing problems and more. This could have a significant impact on the reliability and usefulness of BSS and encourage drivers to return to using their vehicles or choose alternative modes of transport, resulting in increased congestion and emissions and	A BSS optimization framework to minimize the bike rebalancing/ repositioning problems, multi-objective supervised clustering algorithm, mathematical and heuristic approaches.	Pros: 1. The paper suggests ways by which one can recognize the resemblance of bike usage w.r.t. events of time. 2. The paper suggests ways to estimate quantity of bikes parked at a BS station. 3. Bike levels at BS stations have superlative range and this paper

		pollution may increase. This reduces the number of BSS users and reduces system revenue. Operators recognized the imbalance and began building more bike stations closer to each other with the aim of keeping them within a five-minute walk. This extensive and enormous report and aggregation of research paper methodologies introduces a brand new methodology of BSSs. Specifically, the framework consists of two levels: We use fast, online, and incremental learning approaches to predict the number of bikes at stations and balance the system. The target is a structure that eliminates the dynamic BS ploblem of relocation for increasing or decreasing the satisfaction levels of the user by decreasing the demand which are actually unmet in reality. The dissertation by the authors contributes to the field in 5 ways.		provides it. Cons: 1. Scope for improvement in modeling future models.
[15]	Assessing the market potential of electric bicycles and ICT for low carbon school travel: a case study in the Smart City of ÁGUEDA	The authors published this research paper which is actually based on the Portugal based city of Águeda. The intent of this study conducted was to analyze the preferences of the students aged between 15-21 in the context of using e-bikes while going daily to school. It also aimed at assessing their longings and towards ICT attributes. The information for examination of the results was collected in three parts. It comprised of a mobility survey and a SCE.	A stated choice experiment.	Pros: 1. It helped determine which barriers students believe to be the most significant in preventing them from cycling to school and evaluated the precise function of

		The part at the inception was aimed at collecting the responses from the students of their travel preferences about what mode they use, thoughts regarding inclusion of ICT, perceptions barrier for not cycling and previous cycling experiences. This part was named as the "Simplifying Cycling Mobility". The second part dealt with the household budget and business perspectives and in this both the students and their parents were questioned in order to get various insights. This part was named as the "Assessing students and their parents' preferences". In the third part a SC experiment was performed to understand the trade-offs data focussed towards car travel and e-bike relevant attributes by gathering 2232 observations in that regard. The researchers had also taken the final shot at studying the main determinants of both the traditional as well as electric bikes impact on school-to-home/home-to-school trips, using a detailed study of the design using exploratory data analysis and MLRM.		cycling infrastructures. Cons: 1. In the context of student transport in the nation, there is a considerable market innovation potential for cycling that has to be investigated. To do so, a number of barriers of varied degrees of complexity need to be overcome.
[16]	Bike sharing: A review of evidence on impacts and processes of implementation and operation	The authors of this research-paper worked on a very special and distinct idea and approach of reviewing the existing schemes and research-suggestions and findings of previous researches on bike-sharing: how well the schemes of implementation and operation suggested by the past research-approaches have	Surveys, discussion and analysis of various impacts on diverse factors.	Pros: 1. A full summary with all the minute details in the colossal world about escalating quantity of information

		actually come out victorious or to what extent such were actually guiding the working of the present bike-sharing schemes present in different parts of the world actually implementing such schemes. They provided a comprehensive review of such evidences. By following a two-fold measure: understanding and examining gaps in the stuffs that can prove things or simply evidence as well as limitations which require investigation on a farther level; & making more layers on the review of the evidence and justifying whether the positive-sides of the approaches mentioned actually aid in transferability, beneficial impacts and operation-processes or not, they sought to put light on both the impacts as well as the processes (rather than just processes) as a target to enhance the present knowledge reservoir on BS and also contributing towards a lot other stuffs targeting to elevate the measures motivating cycling in the proper way. A sectional segmented approach was taken by the authors, as they sought to provide about escalating quantity of information vents as well as the ever increasing torso of knowledge about BS, an overview that is very very critical of it; discussing many aspects of BS's significance and limitations after summarizing them; providing an evidence from the point of		vents as well as the ever increasing torso of knowledge about BS is shared in this paper. Cons: 1. Many scopes of improvement

		view of process evaluation in managing the deals & operations discussing about BS; they concluded it as they shed light on how the proof shown here can strengthen and transfer positive results to other contexts in terms of impact and implementation processes, and identify key areas for further investigation.		
[17]	Social Media Strategies Used in Marketing Custom Bicycle Framebuilding Companies	The long paper which comprised of more than one examples in the form of case studies sought to traverse utilizing social media, what various master plans the owners of the microenterprise need to sell as well as market. Through the means of semistructured interviews and open-ended questions, data were collected from 5 bicycle framebuilding companies from a south-western US state. The diffusion of innovations theory was used. A thematic analysis identified seven themes from the data, viz., technical proficiency, building a social media presence, effective use of social media platforms, effective communication skills, building a brand identity, time management, and obtaining external support. The overall study findings aimed to help artisan microenterprises learn to use social media effectively, which would lead to boost in sales and profits, further leading to good positive environment for growth and development. The findings also expect their way out towards helping the local economy as it	The diffusion of innovations theory, thematic analysis.	Pros: 1. Helped artisan microenterprises learn to use social media effectively. 2. Boosting sales and profits. Cons: 1. Very antique analysis and processes involved.

		helps to prevent the money from leaving local economies, thus building strong communities.		
[18]	Bicycle sharing systems demand	The study methodology proposed by the authors in this research paper aims to study the demands of the bicycle sharing systems, at other major regions of illustrations including top notch urban demand. The literature survey conducted by the authors of this research paper presented some broad ideas, which included types of bicycle-sharing systems right from the antique to the modern systems; studies involving demand for cycling and BS: latent demand score method, 'revealed' or 'stated' preference surveys; etc. The methodology suggested by the authors to study the demand distributions for bike-sharing involves the two parts: quantifying the other case study demands; and how exactly the characteristics of a trip escalate in demand as various effects are defined upon them. The second part encompassed analyzing the factors such as purpose, distance, slopes, etc. and their effects on the the bicycle sharing demands thus making a cohesiveness study among the various factors proposed. After studying all these, the authors put the proposed methodology and their knowledge to exercise on the case study of Coimbra, Portugal. The conclusion came with pointing out the advantages being that	Latent demand score method, 'revealed' or 'stated' preference surveys.	Pros: 1. Gives a swift test which can be spread to other urban bodies and cities according its characteristics. 2. Bestows a good way for estimating demand of the BS. 3. Permitted to geo-reference the demand, taking into account the features of the city and the visits. Cons: 1. Many socio-economic characteristics not taken into consideration.

		the methodology proposed a valuable technique which can prove to be good in designing the full system. The paper ended with the authors pointing out the areas in which there is a scope for further studies in this area.		
[19]	Are Bikeshare Users Different from Regular Cyclists? A First Look at Short-Term Users, Annual Members, and Area Cyclists in the Washington, DC Region	In this paper, the authors investigated the travel behavior characteristics of bicycle system users. A comparative analysis was done to understand the differences between annual members and short-term user profiles on Capital Bikeshare (CaBi). The data used for the overall research was gathered from a survey of 2007-2008, an online survey of yearly CaBi subscribers and an intercept 10 survey of short-term CaBi users. This paper deals with a case study of BS users (Capital 13 BS) who are short-time and at the same time annual in a couple of regions. The monthly and annual users were subjected to an online membership survey, while others were examined on the basis of 23 survey questions. The long-term members' goals included gathering data on the demographic and usage traits of CaBi members, measuring their happiness with the system, and tracking changes in their travel habits as a result of bikeshare availability. Instructions for completing the survey online were given to short-term users who lacked the time to complete the verbal 19 survey. The analysis part	Surveys and analysis.	Pros: 1. A study that makes suggestions on the differences between cyclists in the Washington, DC, area and short-term CaBi users and members. Cons: 1. The two CaBi surveys' spatial limitations and the possibility of sample-level biases prevented a thorough analysis. 2. Results could not be extrapolated to other US cities from the Washington, DC region.

		comprised of the discussion on various fancy aspects. Finally before concluding the paper, the authors also touched upon some of the scopes for future research which included whether there could be significant differences or not between Area and CaBi users when controlling for other factors, admitting that the analysis cannot adequately account for the two CaBi surveys' spatial limitations, presence of potential biases over the sample. Results for the Washington, DC area may not apply to other US cities, etc.		
[20]	Challenges and Opportunities in Dock-Based Bike-Sharing Rebalancing: A Systematic Review	The approach of how the managing authorities of the bikesharing systems manage the problem of imbalance or bike rebalancing is surprisingly directly related to user-level satisfaction and thus towards profits. The authors of this research paper sought to present a thorough review of the challenges and opportunities in rebalancing of bike-sharing systems (only for 4th generation of BSS). The objective of their research points out towards collecting research papers based on the repositioning-problem in dock-based bikesharing systems, classifying them and to suggest and divert to many novel research venues. A period-wise table containing the main research-topics in BSS research over the decades has been provided in the research paper. Mainly 4 key themes were addressed by the paper.	Various methods of data acquisition, exploratory data analysis.	Pros: 1. Exploratory analysis on researches conducted previously in the field shown. 2. Creation of taxonomy/ classification of proposed algorithms and a good summary of exhaustive discussions provided. Cons: 1. Completely theoretical approach with very less or little emphasis on practicality.

		The methods of data acquisition included performing keyword search in Google Scholar utilizing some selective constraints, and further steps, etc. Contextualising research in the realm of BS was done by performing an exploratory data analysis, by making use of VOSviewer. The research-topics were clustered in 5 groups, and were also further analyzed. Following the major sections of the BSS paper, the authors provided the Summary of Results and Discussion. Additionally, a comprehensive table was supplied to help scholars from other fields address the unresolved issues in the area.		

Chapter 3

Proposed Methodology

3.1 NEED BEHIND THE PROPOSED WORK

Although many researches on bike-share systems have been conducted earlier, on different aspects of the commute methodology utilized in it, even on few broad scenarios as well, case studies focusing upon a particular region or area in which many different factors have been researched upon and analyzed still are not sufficient. Since, it is a broad topic, generalized results/conclusions presented in some of the researches ain't never sufficient for applying to all the regions of the globe.

The BSS Industry no-doubt has huge potential in the future. The below mentionedpoints prove and illustrate the same:

1. The BSS industry has been prophesied to grow at a brawny 5.02 percent CAGR.
2. People across every corner and crevices of the huge world are evolving to extreme level health conscious individuals.
3. There has been an alarming and rapid increase in vigilance on cardiovascular endurance, muscular resistance to fatigue, flexibility, reduced stress levels, strengthening of bones and reduced levels of body fat, increased postural activities & coordination and better-joint mobility.
4. The rapidly increasing urbanization and consequently congestion had raised the need for sustainability due to which the steady rise in BSS need and demand has been noticed in India.
5. For both traditional as well as electric bicycles, the cycling market place of the enormously big world is predicted to be hungry for Rs. 4.4 lakh this year's crore.
6. The govt. has been very very actively trying to promote the healthy and best deed of commuting by a bicycle and because of this it is trying to rollout infrastructural benefits and changes.
7. Impact of COVID-19: The greatly feared virus had restricted large sections of the population to maintain social distancing. So, in order to hit nearby areas like markets, etc. people were seen to avoid public transport and instead use bicycles to commute to places for survival needs. It has also been greatly proved to add to the physical fitness of a living individual and because of this reason people who find it difficult to frequent the gym tend to prefer this activity.
8. The Indian bicycle market has a significant sharehold of the awesomely gigantic world’s bicycle market. It has been found to represent a share of 1% of the world market and 15% by its volume. Hence, the great Indian market is no joke.

3.2 NOVELTY

Apart from all the data analysis and time-series forecasting methods and techniques used, we were able to come up with a novel analysis and prediction method built right from scratch using only some basic yet useful preliminary python libraries.

The novel approach extends from data preprocessing all the way till modelling ways which further includes training the model, saving the best model, evaluating the best model that we got and finally the prediction based upon the former steps. Its an algorithm based upon medium complexity yet robust enough to come up with accurate predictions. The algorithm can be broadly divided into two, specifically the Data Preprocessing part and the Model Activities. For the data preprocessing, we first take the data.csv file as the input and then do the further operations on preprocessing for it to make it fit to be later fed into the TimeSeriesDataset function by aggregating it in hourly format and taking the corresponding top 10 bike-share stations centered data stacked vertically, thus creating new features in the process. We find the new data frame known as time_df in the process. The next sub-algorithm we devised is for the modelling activities which itself is a combination of four major subsections: training, saving, evaluation and prediction. For this technique, we first of all input the time_df dataframe which we got previously by the method we followed while data preprocessing. Before the beginning of the main approach, we did a little exploratory data analysis for seeing how our formatted or preprocessed data actually looked like then. For doing the exploratory data analysis, we took the most popular 10 stations and plotted the mean of the trip_duration in a graphical format as each time-series had different properties. For simplicity, we plotted the first month of every time-series. We noticed that there was no noticeable trend but each time-series had slightly different seasonality and amplitude. We could have further experimented and checked for stationarity, signal decompositions, and so on, but in our case, we focussed on the model-building aspect only. After the small exploratory data analysis, we passed our time_df dataframe to the TimeSeriesDataset format which was very useful since, we were exempted from writing our own Dataloader, we could specify how our algorithm would handle the dataset's features, and we could normalize our dataset with ease as we had the freedom to normalize each time-series individually and also normalization was mandatory because all time sequences differ in magnitude. The trip_duration was the target variable in our case. After that, we found out the actuals and started building and then training our model. In training our model we used the EarlyStopping procedure so that it kicks in everytime the model was on the verge of getting over-trained. After that we fitted our model, and also loaded and saved the best model. For the evaluation of our model we calculated the average p50 loss by comparing the actuals and the predictions. We finally made the predictions and created one plot for each to_station between the actual and the predicted so as to get a good picture of our study.

Algorithm: Data Preprocessing

Input: *df* = data in ".csv" format

Output: *time_df* = {data for (aggregated in Hrly. frmt.) + (values for top 10 "*to_stations*") stacked vertically}

```
1: procedure PREPROCESSDATA(df)
2:     df = df.resample('1h').mean( )
3:     df = df [['ST_01' to 'ST_10']]
4:     df_list = [ ]
5:     for all label in df do
6:         ts = df [label]
7:         start_date = min (tf)
8:         end_date = max (tf)
9:         active_range ← (ts.index >= start_date) & (ts.index <= end_date)
10:        ts = ts [active_range]
11:        tmp = {'trip_duration' : ts}
12:        extraction : tmp['hours_from_start'] , tmp['days_from_start'] ,
           tmp['date'] , tmp['to_station'] , tmp['hour'] , tmp['day'] ,
           tmp['day_of_week'] , tmp['month']
13:        time_df = df_list.append(tmp)
14:    return (time_df)
```

Fig. 1. Data Preprocessing algorithm for novel approach.

Algorithm: Model Activities (Training, Saving, Evaluation, Prediction)

Input: *time_df*

Output: *best_mode_path, predictions*

1: **procedure** MODELACTIVITIES(*time_df*)

2: training = TimeSeriesDataset(*time_df*)

3: validation = TimeSeriesDataset.from_dataset(training, *time_df*, predict=True, stop_randomization=True)

4: train_dataloader = training.to_dataloader( )

5: val_dataloader = validation.to_dataloader( )

6: actuals = torch.cat([y for x, (y, weight) in iter(val_dataloader)])

7: baseline_predictions = Baseline( ).predict(val_dataloader) (actuals - baseline_predictions).abs( ).mean( ).item( )

8: ***initializations*** **:** early_stop_callback = EarlyStopping( ), lr_logger = LearningRateMonitor( ), logger = TensorBoardLogger( )

9: trainer = trainer.fit( )

10: save(*best_model_path*)

11: load(*best_model_path*)

12: *predictions* = model.predict(val_dataloader)

13: print{loss = (actuals – *predictions*)}

14: **for all** cus **in** range **do**

15: plot(cus, graph)

16: return (*best_model, predictions*)

Fig. 2. Algorithm of Modelling Activities for the novel approach.

3.3 DATASET USED

The methodology of the entire process of achieving the goal of devising target specific marketing strategies towards converting the customers of the Divvy's bike share system towards dedicated subscribers for enhancing and boosting the profits of the bike share company involves a multitude of steps, procedures, approaches, thoughts and implementation techniques. The Chicago Data Portal serves as the formal website from where all the data which is used to perform all the analysis is gathered. It is officially an ocean or enormous locker which takes the headache and responsibility of keeping government data in one piece and also to provide the access to all the valuable data which it has in store for motivating all the developers to serve the diverse community of the great place called Chicago by creating creative and greatly innovative tools. The site is managed by none other than the great government. It has a lot of datasets (over 600 to be precise) ranging from facilities to city departments to services to performance in formats which can be used by even a child. The data of our interest is Divvy's bike sharing trips information. Divvy is a well-known and ubiquitous name and is well reputed to provide bike sharing facilities across Chicago and Evanston. Exploring Chicago was never fun until the onset of Chicagoland's BSS which gives the convenience to both the visitors as well as residents an affordable alternative in a fun as well as interactive way to explore the great land and also to have the once in a lifetime pleasure and opportunity of getting around the enchanting and fabulous land. Divvy is very popular to have its robust as well as massively strong durable bikes. Those bikes are build only on special purpose. Throughout the enchanting and marvellous land the custom built bikes of the company are locked in a special way into a network of stations which are used for docking purposes. Those bikes allow the flexibility to be purposely locked as well as unlocked in different stations across the special region. Those colossal fortunate individuals in the region use the BSSs to do a multitude of activities some of which include going to appointments, to work or school and also to do small part-time jobs. The services are available 24x7 which means access to all the stations as well as bikes in the whole massive system, for the users. All the data as well as metadata for each and every trip gets listed for each and every trip in the dataset which is famously known by the name of Divvy Trips. At the time of analysis, the dataset consists of 21,242,740 rows, containing valuable and extensive information about each of the bike trips from 2014 to 2019. The dataset is an example of a living dataset as regular updation adds more and more data to it, thus adding to the bulk of knowledge. The data contains info about the Trip Id, Start Time, Stop Time, Bike Id, etc., of each of the bike sharing trips monitored by the company. The data has been made public by the company for public use for policy makers, transportation professionals, web developers, data analysts, and a lot of other people to use for maintaining their workflows. The data about each of the trips provided by Divvy is anonymous. The data is processed to remove trips made by staff during system maintenance and inspections. Rides under 60 seconds. This can result in false starts or the user attempting to redock the bike to ensure safety. The data is publicly available for download in a static format, options like CSV, JSON, etc. are available.

3.4 ALGORITHM USED

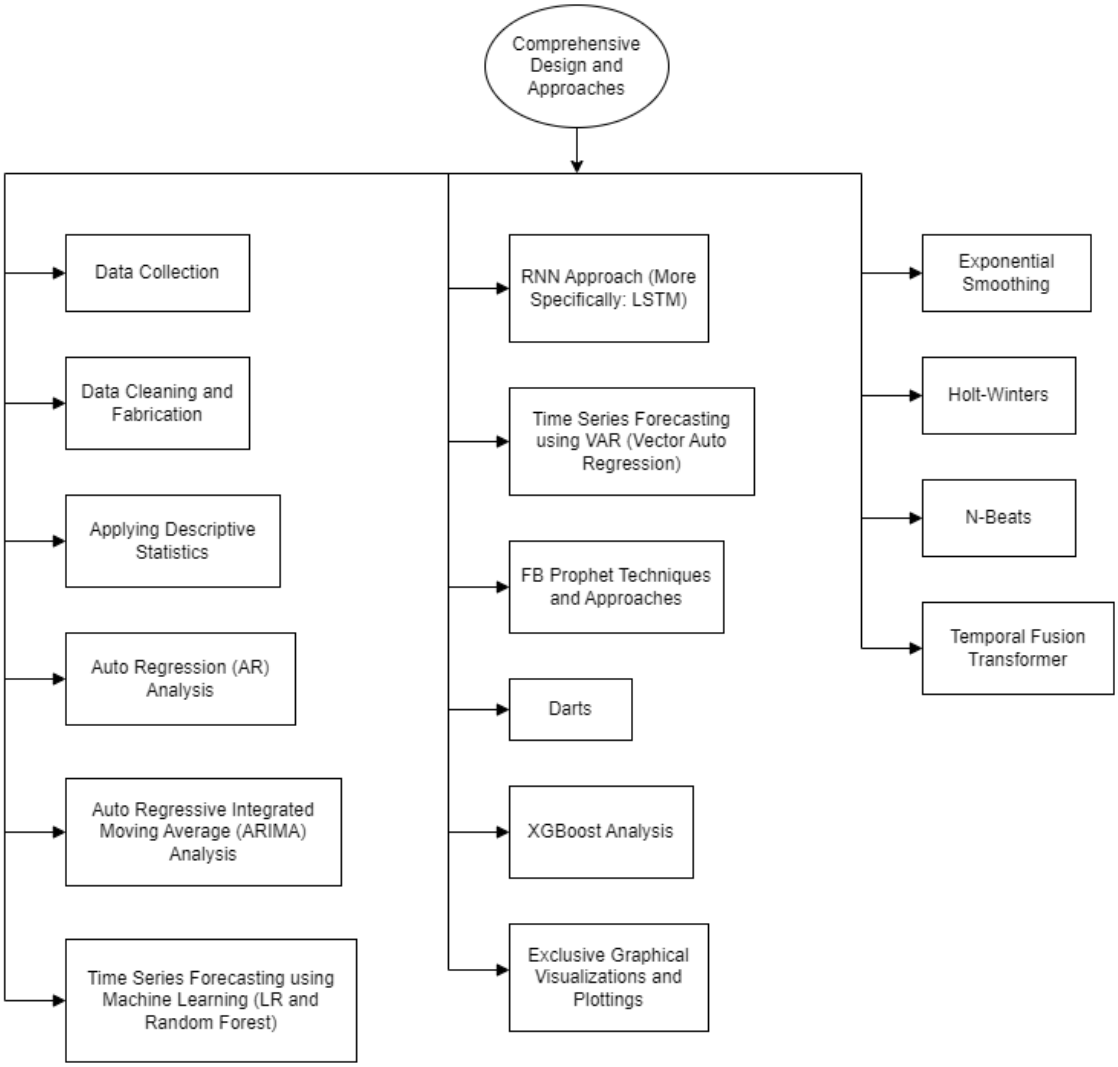


Fig. 3. Proposed system diagram consisting of all the major algorithms used.

As mentioned before, the process starts with searching for and downloading the dataset of Divvy Trips from the Chicago Data Portal, including it in our workflow if all the contraints comply. The ROCCC approach is used to determine the credibility of the data. It demands the data be: (i) Reliable - It is complete, accurate, and represents all bike rides made in the city of Chicago during the selected analysis period. (ii) Original - The data is provided by Divvy, which operates the city of Chicago's bike-sharing service Divvy. (iii) Comprehensive - The data includes all ride details such as start time, end time, station name, station ID and membership type. (iv) Current - It is up-to-date. (v) Cited - Data is quoted and available under a data license agreement. The coding and data analysis is done predominantly in R and Python. The first step involves the importing of data, cleaning the data, and sampling the data as we have limited computing power and our system cannot process over 21 Million data tuples all at once. Hence, to gain a basic understanding first, sampling of the data is necessary. For data cleaning we use R Language. We first loaded all the necessary packages for doing so, followed by sampling.

We added new columns: Date, Year, Month, Day and Day of the Week. We then checked the data for errors. We cleaned the column names and checked for duplicate records in rows. The exporting of the cleaned data to a new file is followed by updating the working directory to the script path. After that, we started the descriptive analysis, in which saw the average ride time by each day for Subscribers vs Customers. We sorted the days of the week, analyzed the ridership data by type and weekday, and visualized the quantity of rides based upon the type of the rider, and also visualized the average duration of ride by rider type.

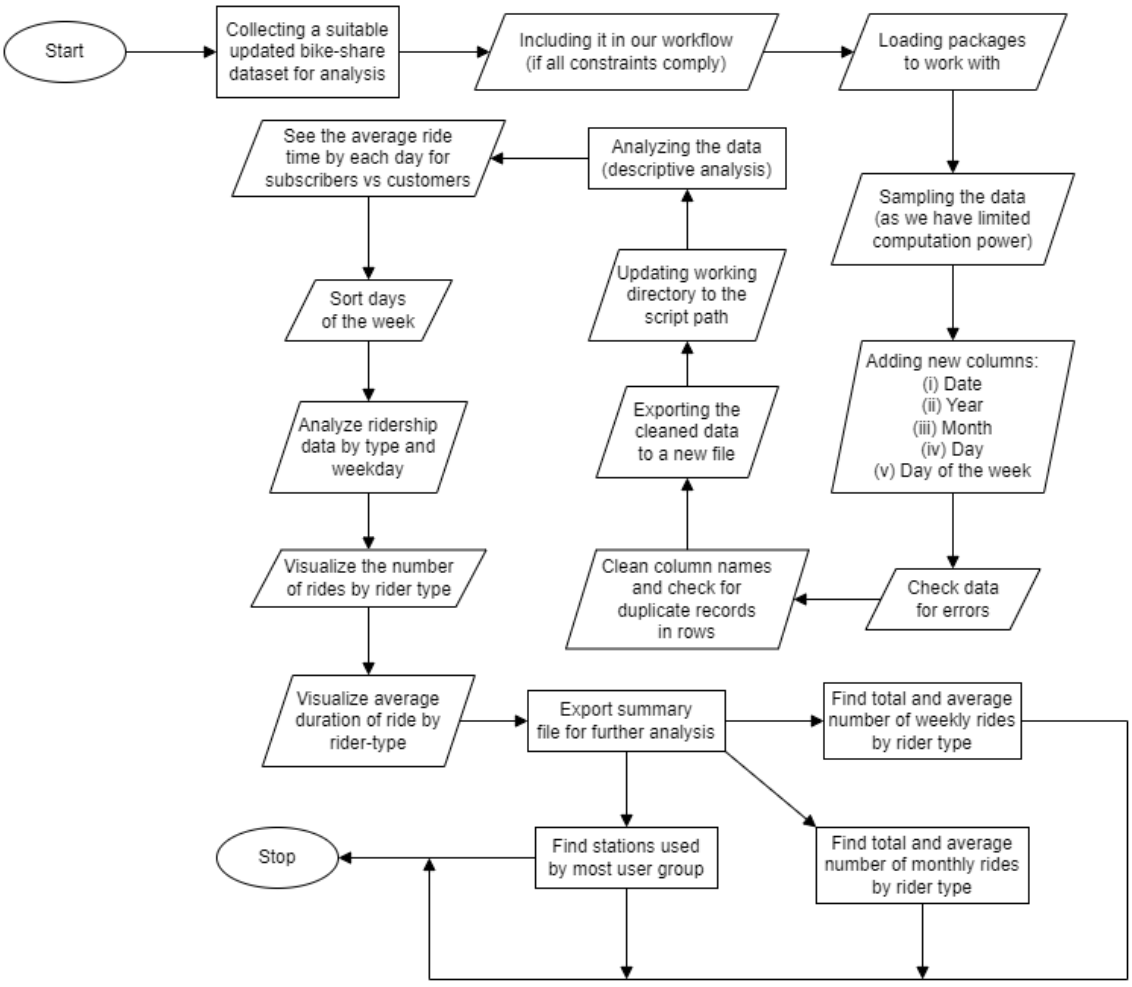


Fig. 4. Combined algorithm for Data Collection, Data Cleaning and Descriptive Analysis of the data.

We also performed a number of statistical techniques to know some more insights into the data that we were working on which included various machine learning techniques, statistical tests, visualization procedures and everything one could do to figure out the various parameters hidden in the dataset, the trends which it indicated and whatever it wanted to convey overall. We also trained and tested the data many times, each time with a different technique and approach and predicted the outcomes. Suitable

metrics to validate the accuracy of each of the statistical schemes like MSE, RMSE, etc., were used, and whatever seemed justifiable according to the credibility of the situation.

Fig. 5. Graphical results obtained from Descriptive Analysis of the dataset.

An Auto Regression (AR) analysis was performed on the dataset. In AR, we have some inputs and we multiply it with some weights and get the outputs as a continuous value. It is called Auto Regression because here the inputs are the previous values of the time series. In this analysis, we had tried to predict the future trip durations or the trip durations beyond the sample of data which we have actually taken into consideration. The inputs are the past trip durations, each of which came with the start time of the bike share trip. The trip durations corresponding to the actual date and original time stamps consisted of the inputs. So, the entire process starts with importing the necessary python modules which we used for analysis like numpy, pandas, etc. In addition, we also imported the AutoReg module from statsmodels for the Auto Regression part. After all these required inclusions, we read the previously cleansed dataset and extracted only the necessary parameter i.e. the trip duration along with the timestamps for our analysis, and also printed the descriptive info about the data for our convenience. We then checked our data for stationarity. Our data needed to be stationary for us to get a good time-series prediction. A stationary data means that the statistical properties should be constant, viz., mean should not change, there should be no change in variance, and there should be no seasonality or repeating patterns or trends in the data. After plotting the data, and making it certain that we were not sure whether the data is stationary or not via the method of visual inspection, we performed the ADF Test to check for stationarity of the data. After performing the ADF Test, we got some values which included ADF value, P-Value, Number of Lags, Critical Values, etc. For making the conclusion about the data being stationary or not we inspected the P-Value. The P-Value is a kind of probability value on whose value we can actually get the null hypothesis to be accepted or rejected for the good. As the P-Value for our case was less than 0.05 (statistically significant), we were able to reject the null hypothesis, by which we concluded that the data was stationary. Then, for training and testing purposes, we set aside a portion of the data for model training and a smaller portion for testing. After splitting the dataset into training and testing set, we fed the training lot onto the AutoReg function specifying the no. of lags and then fitted the model. After training the model, we checked the parameters of the model and its summary. We got a lot of values after the training process like Log

Likelihood, AIC, BIC, HQIC Scores, etc. We noticed that the P-Values were not less, which indicated that the time lags were not that significant. We then tried to make predictions on the test set by specifying the starting and ending index and do the comparison on the basis of our action of training the model. After we were done with making the predictions, we plotted the results in graphical format. We observed that the predictions were not that bad. After making the predictions and presenting our finding in the form of a graph, were calculted the error in our predictions by making use of RMSE (Root Means Squared Error) value. Finally, we also made future predictions on the trip data and observed our findings for the next timeframe.

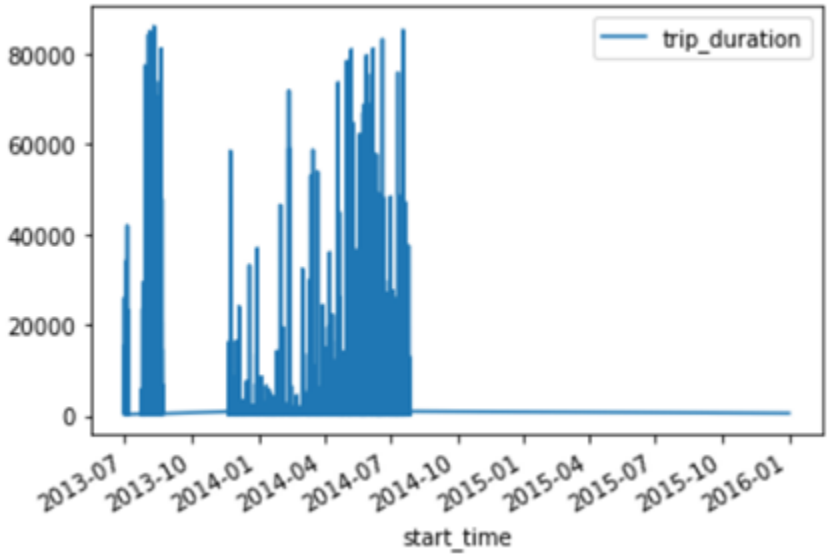


Fig. 6. Plot of start_time vs trip_duration.

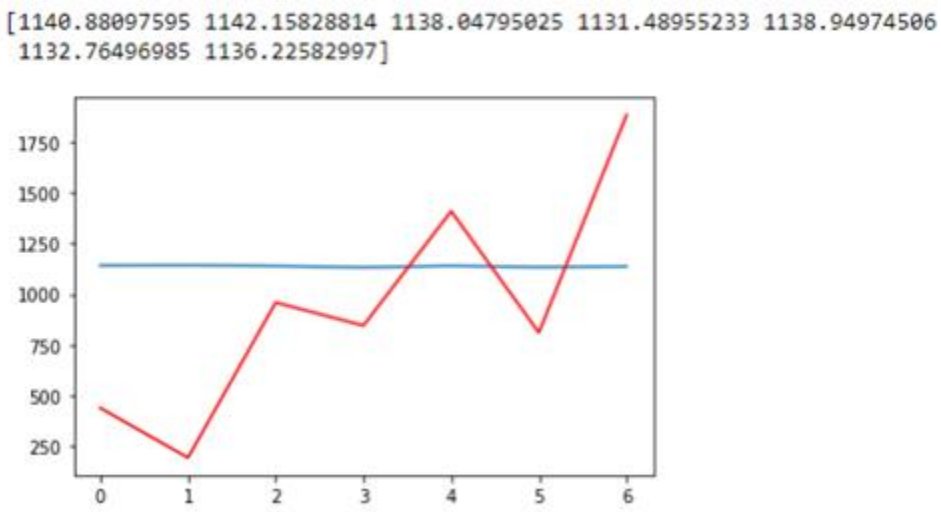


Fig. 7. Plot of the Auto-Regressive test set prediction results.

An Auto Regressive Integrated Moving Average (ARIMA) analysis was performed on the dataset. An ARIMA model is quite similar to ARMA (Auto Regressive Moving Average) model. In ARMA, we take into consideration the previous values of our data in addition to the previous errors which becomes our inputs for reaching the output or future values. The ACF as well as the PACF have a role to play in ARMA. The only difference in ARIMA is the order of differentiating to get stationary values (in addition to ACF and PACF). Hence, in ARIMA, we just convert non-stationary series to stationary series before proceeding. The inputs are the past trip durations, each of which came with the start time of the bike share trip. The trip durations corresponding to the actual date and original time stamps consisted of the inputs. So, the entire process starts with importing the necessary python modules which we used for analysis like numpy, pandas, etc. In addition, we also imported the pmdarima module for the ARIMA part. After all these required inclusions, we read the previously cleansed dataset and extracted only the necessary parameter i.e. the trip duration along with the timestamps for our analysis, and also printed the descriptive info about the data for our convenience. We then ran the ADF-Test as we had done in the previous case. After plotting the data, and making it certain that we were not sure whether the data is stationary or not via the method of visual inspection, we performed the ADF Test to check the data stationarity. After performing the ADF Test, we got some values which included ADF value, P-Value, Number of Lags, Critical Values, etc. For making the conclusion for the data being stationary or non-stationary we inspected the P-Value. The P-Value is a kind of probability value on whose value we can actually get the null hypothesis to be accepted or be rejected for good in the globe. As the P-Value for our case was less than 0.05 (statistically significant), as a result of our ability to reject the null hypothesis, we deduced that the data were stationary. Then, we tried to figure out the order for our ARIMA model (as ARIMA requires 3 parameters viz., PACF, ACF, and the order of differencing as mentioned before). So, from pmdarima we imported the auto_arima function for figuring out the order. The auto_arima function tries to use different combinations for figuring out the order for the model, and for every model it assigns a score, which is called the AIC (Akaike's Information Criterion), and the goal is to minimize the AIC. After getting the best model i.e. the model with the minimum AIC, we splitted the dataset into training and testing as we did before. For fitting the model, we called the ARIMA function which we imported from statsmodels python package and fed the training part onto it after specifying the order. We then tried to make predictions on the test set by specifying the starting and ending index and do the comparison on the basis of our action of training the model. After we were done with making the predictions, we plotted the results in graphical format. After making the predictions and presenting our finding in the form of a graph, were calculated the error in our predictions by making use of RMSE (Root Means Squared Error) value. We observed that the RMSE value is very different and is actually quite more than the Mean value, which made us conclude the model to be bad one. After retraining the model on the entire dataset, we tried to make the future predictions.

Figure out Order for ARIMA Model

In [7]:
```
from pmdarima import auto_arima
# ignore harmless warnings
import warnings
warnings.filterwarnings("ignore")
```

In [8]:
```
# calling the auto_arima() function
# Auto ARIMA is going to try for different combinations (like (0,1,0), (0,1,1), (0,1,2), etc.)
# and for every order it is going to assign a score called the AIC
# the goal is to minimize the AIC (Akaike's Information Criterion)

stepwise_fit=auto_arima(df2['trip_duration'],trace=True,suppress_warnings=True)

# performing step-wise search to minimize AIC score

stepwise_fit.summary()
```

Fig. 8. Code snippet for figuring out the order for the ARIMA model.

Time series forecasting using machine learning was performed on the dataset. Two ML models namely Random Forest and Linear Regression were used to make predictions. We read the previously cleansed dataset and extracted only the necessary parameter i.e. the Trip Duration along with the timestamps for our analysis, and also printed the descriptive info about the data for our convenience. We sorted the data on the basis of timestamps. After that, we plotted the data in the form of a graph. In supervised learning, we need an input and an output for our model to learn the relationship between the input and output to implement our ML models. Hence, we created 3 additional columns in our dataset by shifting the values from the original trip_duration column by one row, successively for each of the successive columns. By doing this, we ensured that we make the shifted values as the outputs for the original inputs, thus solving our problem of fetching the outputs for training our models. We imported functions like LinearRegression and RandomForestRegressor from sklearn package for our further work. After a slight data preprocessing, and dividing the dataset into the lots utilized for training and testing utilities, we fitted the 2 models onto the training data. After doing these operations, we predicted the results by using the test set for both Random Forest and Linear Regression as well and plotted the results in 2 separate graphs, one for each of the models. We observe that the models were capturing the trends quite well. To compare the accuracy of the predictions made by the two models, we calculated the MSE (Mean Squared Error) for each. The MSE for Random Forest Model came out to be around 1382.16, whereas that for Linear Regression was around 1328.63, which were not that different. Hence, we concluded that the Linear Regression model was slightly better that the Random Forest model, although the deviation was very insignificant.

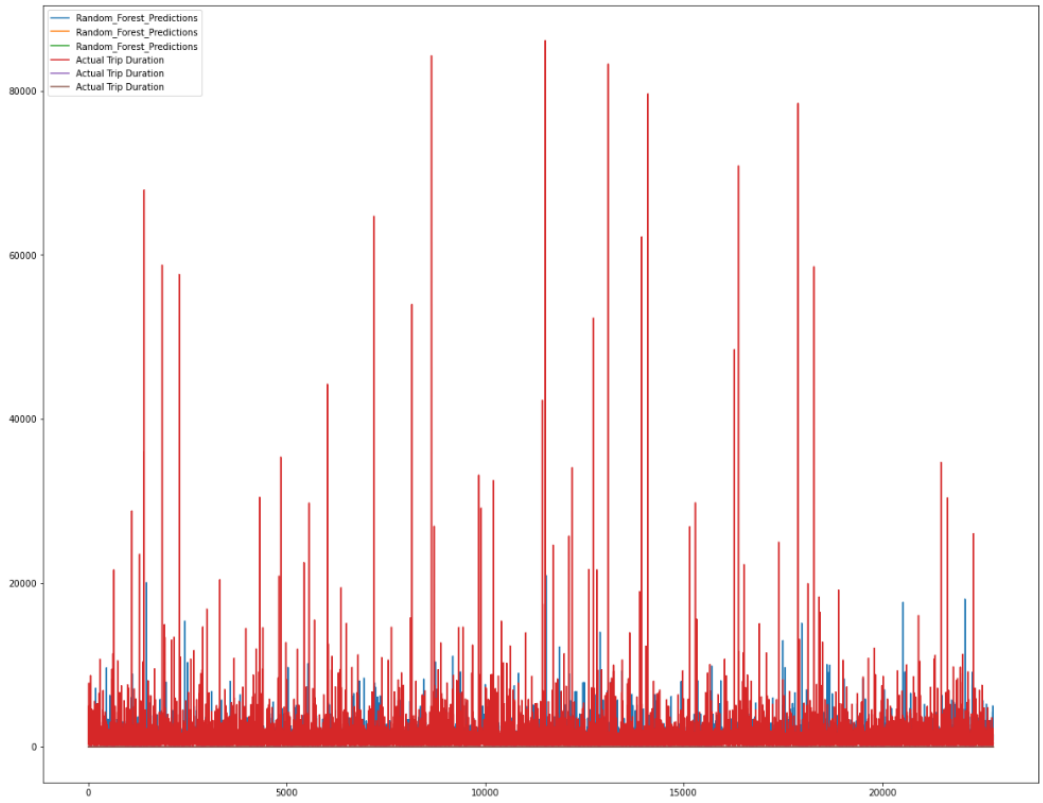


Fig. 9. Graphical plot for the Random Forest predictions.

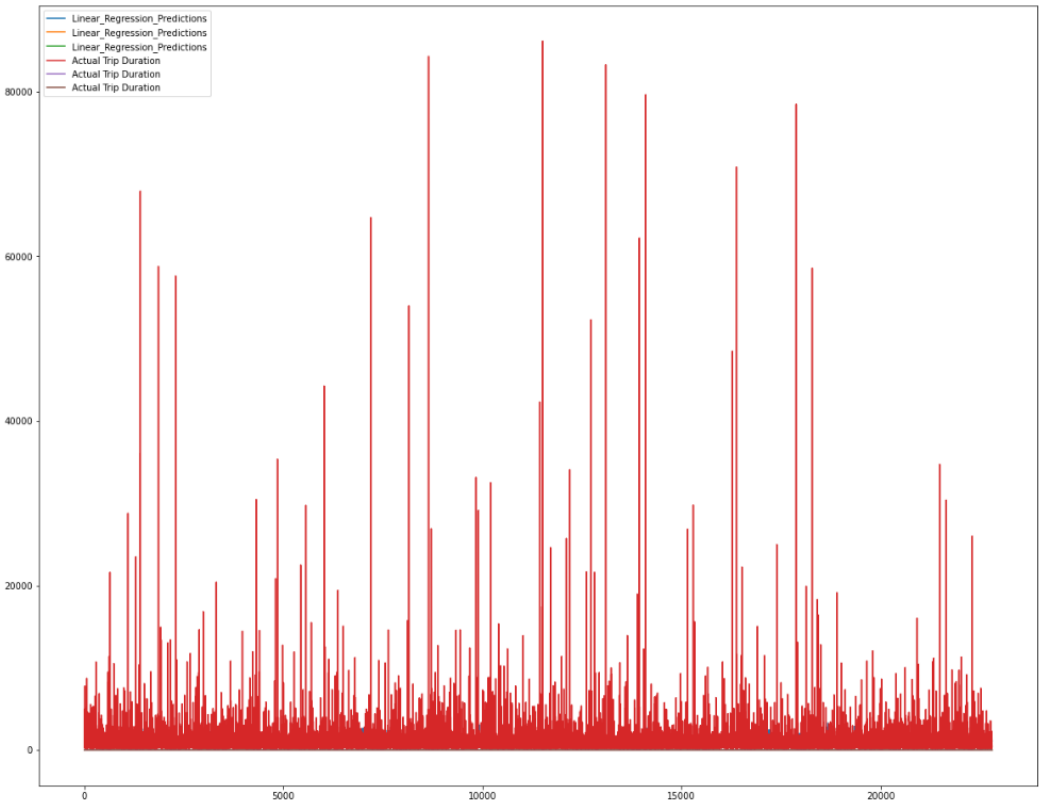


Fig. 10. Graphical plot for the Linear Regression predictions.

A RNN (Recurrent Neural Network) approach or more specifically LSTM (Long Short Term Memory) was also applied to the dataset. A RNN is a form of ANN that is commonly used to make predictions on data where there normally involves a sequence. LSTM is a form of RNN. So, the entire process starts with importing the necessary python modules which we used for analysis like numpy, pandas, matplotlib, etc. We read the previously cleansed dataset and extracted only the necessary parameter i.e. the Trip Duration along with the timestamps for our analysis, and also printed the descriptive info about the data for our convenience. We sorted the data on the basis of timestamps. After that, we plotted the data in the form of a graph for our convenience. To view the exact seasonal nature and how the seasonality in our data actually looked like, we imported a function called seasonal_decompose from statsmodels package. We hadn't checked for the stationarity in our data, because technically RNN can work with non-stationary data as well and do not require the data to be stationary. A training set and a testing set were then created from the dataset. We preprocessed the data using a MinMaxScaler for converting the data into a scale of 0 to 1. For doing so, we first fitted the training set using the scaler object, and then we transformed both the training and testing sets using the transform function of the scaler object. We then formatted the data for feeding it to the neural network model. For doing so, we define the generator which consists of specifying the no. of inputs (assigning 3 to it), no. of features (assigning 1 to it). The no. of features would have been more if we were dealing with more than one time-series, but that is not the case with us here. Then we called the TimeSeriesGenerator function and give it the scaled trained input, the no. of inputs, etc. After calling the Sequential, Dense, and LSTM classes from Keras, we successfully generated the model. A 100-neuron LSTM layer was added, with the RELU activation function serving as the layer's activation function. Finally, we built the model with MSE as the loss function and the Adam optimizer. We fitted the model for 50 epochs. After training the model, we computed the loss for each epoch, and we also plotted it. We took the last 12 values in the training set to predict the first value in the test set. We observe that the original value was 0.01374463 whereas the model had predicted 0.01540671 which is pretty close. Then we made predictions on the testing set and converted it back into the original scale and performed the Inverse Transform upon the test predictions and appended those predictions to our original test set, and finally, plotted those two in the form of a graph to see how similar or different they are from each other. To put a number to see how good the predictions were we used RMSE which came out to be something around 1268.4964.

```
In [40]: test_predictions=[]

first_eval_batch=scaled_train[-n_input:]
current_batch=first_eval_batch.reshape((1,n_input,n_features))

for i in range(len(test)):

    # get the prediction value for the first batch
    current_pred=model.predict(current_batch)[0]

    # append the prediction into the array
    test_predictions.append(current_pred)

    # use the prediction to update the batch and remove the first value
    current_batch=np.append(current_batch[:,1:,:],[[current_pred]],axis=1)
```

Fig. 11. Code snippet for calculating the test predictions for LSTM implementation.

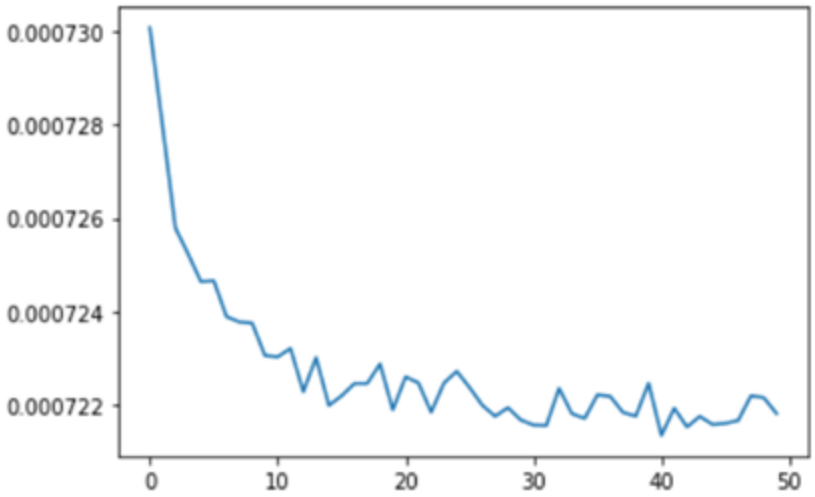


Fig. 12. Graphical plot for loss per epoch while fitting the LSTM model.

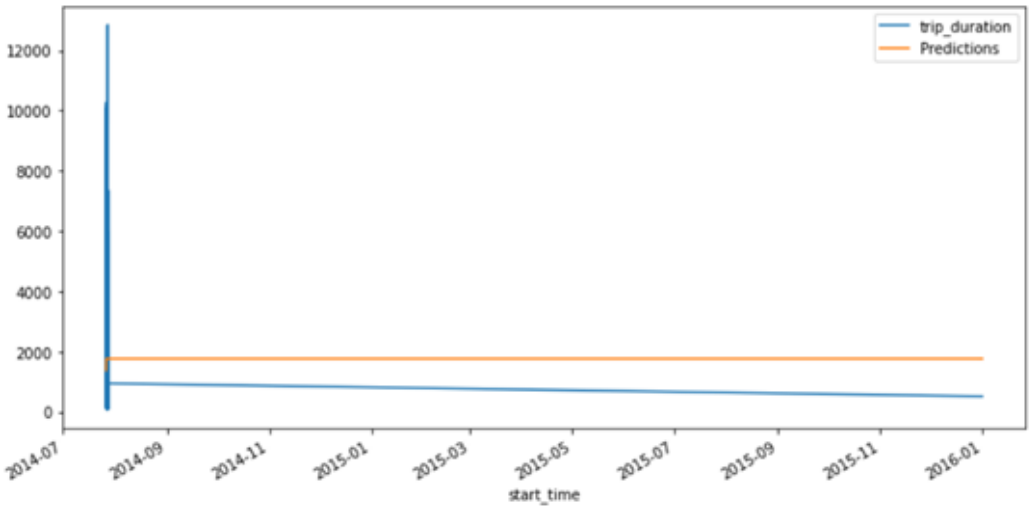


Fig. 13. Predictions graph for trip duration from LSTM implementation.

Time series forecasting using VAR (Vector Auto Regression) was performed on the dataset. Normally, in Auto Regression we predict the future values based on the previous values or the previous time lags of a particular time series to make predictions into the future. In VAR we assume that there are two time series which have a correlation. We started by importing the cleansed dataset and this time in addition to the Trip Duration, we also included some other parameters like From Station Id, To Station Id, From Latitude, From Longitude, etc. We also imported various classes and functions like plot_acf, plot_pacf, VARMAX, VAR, grangercausalitytests, dfuller, etc., mainly from statsmodels package and others like tqdm, itertools, etc. And, as always we also imported python modules like numpy, pandas, matplotlib, etc. We plotted the 8 parameters into 8 different graphs. We checked for stationarity for all the 8 time-series using ADF Test. We inferenced after observing all the P-Values corresponding to each of the time series that they all are stationary as all of them give P-value less than 0.05.

Before progressing, we checked whether there was any correlation between 2 suitable parameters or columns of the dataset, because as mentioned before it is the fundamental concept behind Vector Auto Regression. We did that through Granger Causality Test. In accordance with that test, if the p-value is less than 0.05, the hypothesis is valid. We concluded that from station id causes trip duration and to station id also causes trip duration. After all that, we bifurcated the dataset to become two different souls namely the training and the testing lots. To ascertain the number of lags, we fed the difference data of the training set into the VAR function to make the model. After specifying the maximum lags in the select_order function of the model, we observed the summary of our analysis. The results came out in the form of 4 scores namely AIC, BIC, FPE and HQIC scores. For a model to be good, these scores must be as low as possible. For AIC the minimum score was observed to be 1.651 in the lag no. 81, minimum score of 1.929 for BIC in lag 29, 5.211 for FPE in same lag as that of AIC, and finally for HQIC, the minimum score of 1.773 was found to be in lag 45. To fit the model we used the VARMAX class in which we fed the training data. One advantage of using that class was that it is known to make forecasting very easy. After the computation, we got the AIC score to be 1473817.706, BIC as 1474584.888, and HQIC as 1474054.573, and we also got other informations such as results for equation trip_duration, results for equation from_station_id, results for equation to_station_id, error covariance matrix, etc. For making the forecasts from the point where the training data ends, we specified the number of forecasts as 7600, and also calculated the mean of all the predictions. Finally, we calculated the mean values of each of the time series values and the RMSE for comparing the errors corresponding to each series. We observed that the RMSE of trip duration is more than its mean, which is statistically not good. Although the same is the case for from longitude and to longitude as well, but it is because of the negative natures of its values and hence including it makes no sense. For other parameters, the RMSE values are quite less than the respective mean values, which meant that it was statistically good.

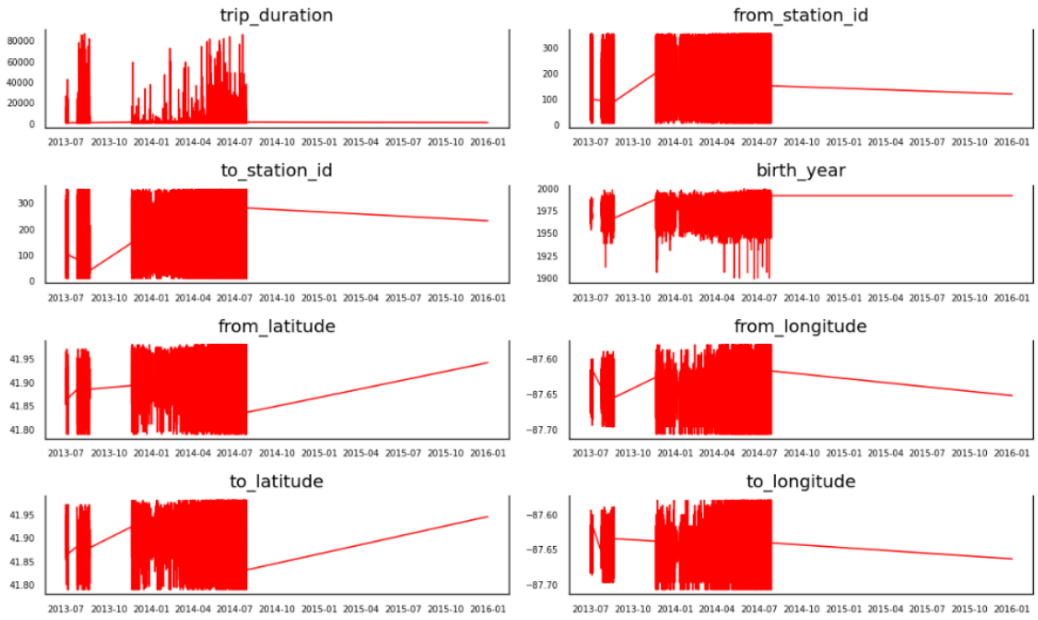


Fig. 14. Tight plot layout of 8 attributes vs start time from the dataset.

The Core Team in the domain of Data Science at the tech company which is known by the name of Facebook developed a software which was open-source known by the famous and marvellous name of Prophet. The software package can be downloaded free of any penny on CRAN and PyPI. An additive model is greatly utilized by the software package called Prophet to tightly fit the non-linear trends that too with seasonality in a total of three types which include yearly, weekly as well as daily particulars. Its heartwarming to mention that it also takes into account the holiday effects. The underlying algorithm as mentioned before, the generalized additive model can be decomposed into three main components viz. trend, seasonality and holidays. Its worth mentioning that both trend and seasonality are two very crucial but those components are very very tough to quantify in a time series analysis. Surprisingly, FB Prophet does deliver a fabulous performance in capturing both of them. FB Prophet can be called a specialist as its well suited for time series having strong seasonal effects. Its also well suited for time series with historical data spanning over multiple seasons. The plus point of FB Prophet is that it does not breaks down whenever it encounters missing data or even changing trends. It also handles the outliers with a tremendous amount of grace.

```
In [2]: import pandas as pd
        from prophet.plot import plot_plotly,plot_components_plotly
        from prophet import Prophet

In [3]: # import csv dataset
        df=pd.read_csv('divvy-tripdata_cleaned.csv')

        # drop the na values
        df.dropna(inplace=True)
        df.reset_index(drop=True,inplace=True)
```

Fig. 15. Code snippet for package importing and data reading in FB Prophet forecasting.

```
In [6]: df=df[["start_time","trip_duration"]]
        df.head()
```

Out[6]:

	start_time	trip_duration
0	07/14/2014 05:17:00 PM	277
1	06/19/2014 06:04:00 PM	566
2	07/09/2014 07:46:00 PM	637
3	06/27/2014 04:19:00 PM	161
4	02/26/2014 07:28:00 PM	210

Fig. 16. Code snippet for showing start time and corresponding trip duration.

Change Column Names for FB Prophet

```
In [7]: df.columns=['ds','y']

In [8]: df['ds']=pd.to_datetime(df['ds'])
        df=df.sort_values("ds")
        df.tail()
```

Out[8]:

	ds	y
19912	2014-07-27 19:07:00	665
25598	2014-07-27 19:17:00	183
148	2014-07-27 19:18:00	1150
42319	2014-07-27 19:19:00	951
3231	2015-12-31 17:35:00	521

Fig. 17. Code snippet exhibiting the replacement of attributes of interest with ds and y in FB Prophet forecasting technique.

```
In [9]: # plot the dataset
        df.plot(x='ds',y='y',figsize=(18,12))
```

Fig. 18. Code snippet for plotting the dataset in FB Prophet forecasting.

Train, Test Split

```
In [11]: train=df.iloc[:len(df)-472]
         test=df.iloc[len(df)-472:]
```

Fig. 19. Code snippet for splitting the data between training and testing sets.

Start Making Predictions

```
In [12]: m=Prophet()
         m.fit(train)
         # making daily predictions
         future=m.make_future_dataframe(periods=207,freq="D")
         forecast=m.predict(future)
```

Fig. 20. Code snippet for making predictions in FB Prophet forecasting.

```
In [13]: forecast.tail()
```

Out[13]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	daily
39588	2015-02-14 13:10:00	849.717445	-877.723277	2561.838850	387.762143	1322.063831	37.073657	37.073657	37.073657	-33.361508
39589	2015-02-15 13:10:00	850.094168	-759.310288	2643.808433	382.885594	1324.660492	57.478614	57.478614	57.478614	-33.361508
39590	2015-02-16 13:10:00	850.470890	-867.498784	2609.175552	377.898944	1326.560116	-56.261059	-56.261059	-56.261059	-33.361508
39591	2015-02-17 13:10:00	850.847612	-1032.077698	2586.746672	372.679401	1329.850814	-52.470053	-52.470053	-52.470053	-33.361508
39592	2015-02-18 13:10:00	851.224334	-817.890385	2599.409994	369.755401	1333.514931	-77.659031	-77.659031	-77.659031	-33.361508

Fig. 21. Last 5 rows of the forecasted parameters in FB Prophet forecasting.

We began by importing the necessary libraries such as pandas and prophet. Then we loaded the cleansed divvy tripdata dataset, dropped the NA values and resetted the index. Then we extracted the Start Time and the Trip Duration columns because we had to make predictions concerning these two columns only and discarded everything else. After that, we changed the columns names to 'ds' and 'y' respectively (The start_time to ds and the trip_duration to y). We had to do that because in order to use the functionalities its essential to do so, as the team at Facebook had hard coded it to make it work in that way. We parsed the start_time to datetime format for future convenience. We splitted the dataset into training and testing sets and started making predictions. For making the predictions, we first invoked the Prophet class and fillted our model onto the training set, and to make the predictions we used the make_future_dataframe function and also specified the periods i.e. the number of time steps into the future for which we were going to make the predictions, and also the frequency. After all that, we made the predictions using the predict function and saved the predictions into the forecast dataframe. After making the predictions, we observed the values like yhat, yhat_lower and yhat_upper, which were actually our values of interest to make the plots. Then, using the built-in FB Prophet visualizations through the plot_plotly function after feeding the

model and the forecasted values we obtained the great visualization. We got an interactive plot where we had the option to set the plot to display as per 1-week data, 1-month data, 6-months data, 1-year data, and the whole trend as well. Also we could see a more detailed fraction of the entire plot by setting the starting and the ending timestamps manually on the figure window. Another visualization which the FB Prophet provides is the visualize components using the plot_components_plotly function. It gives the general trend throughout the dataframe, and also the yearly and the weekly trends. Finally after making the predictions, we had to evaluate how good the model was. In order to do so, we created a separate test set and calculated the RMSE between the actual test set and the predicted values. We also calculated the mean value of the test dataset. The RMSE was around 611.35 and the mean value was around 828.21 which indicated that the model was pretty good as the RMSE was less than the Mean.

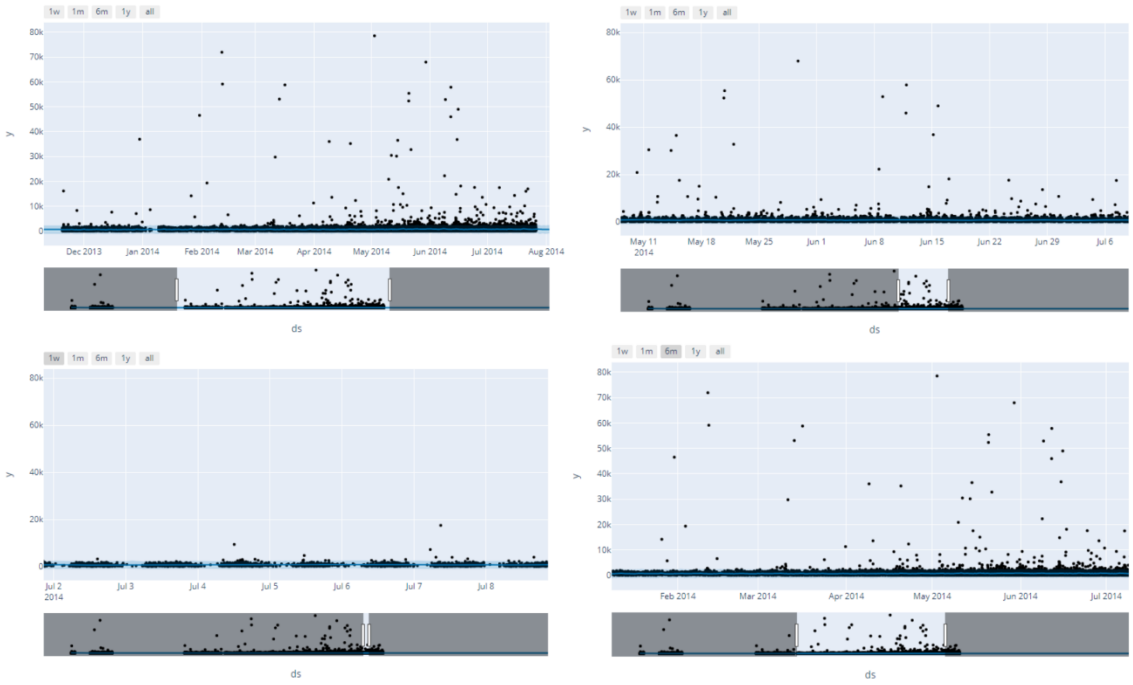


Fig. 22. Interactive plots of the forecasted lots obtained by FB Prophet.

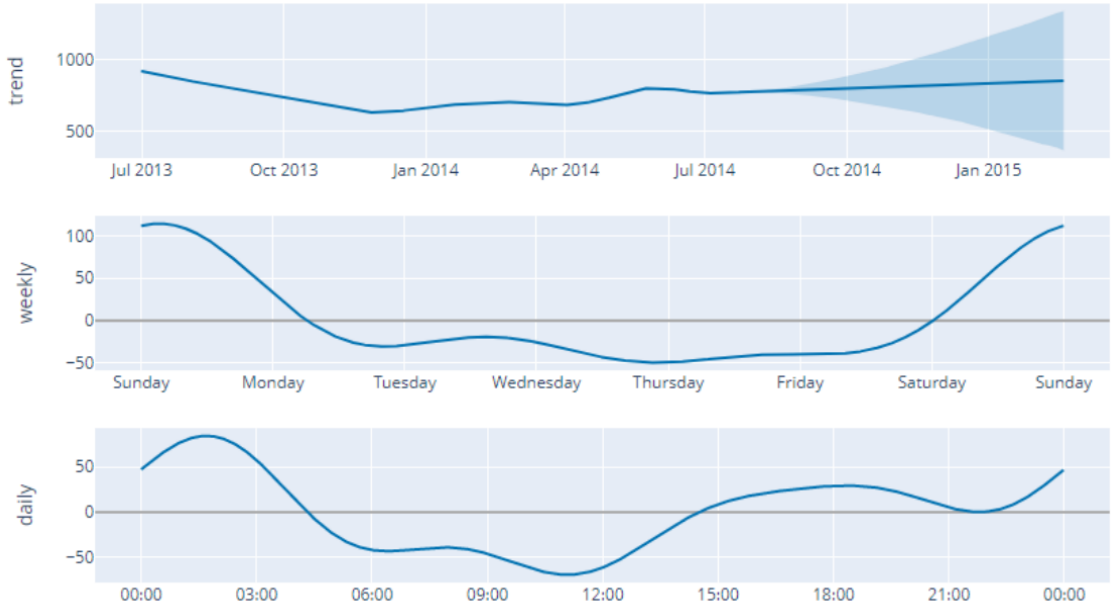


Fig. 23. Graphical visualizations of the various trends obtained using FB Prophet.

Evaluating the Model

```
In [20]: from statsmodels.tools.eval_measures import rmse

In [21]: predictions=forecast.iloc[-472:]['yhat']

In [24]: print("Root Mean Squared Error between actual and predicted values: ",rmse(predictions,test['y']))
         print("Mean Value of Test Dataset: ",test['y'].mean())

         Root Mean Squared Error between actual and predicted values:  611.354376769347
         Mean Value of Test Dataset:  828.2097457627119
```

Fig. 24. Code snippet for the evaluation of the FB Prophet model.

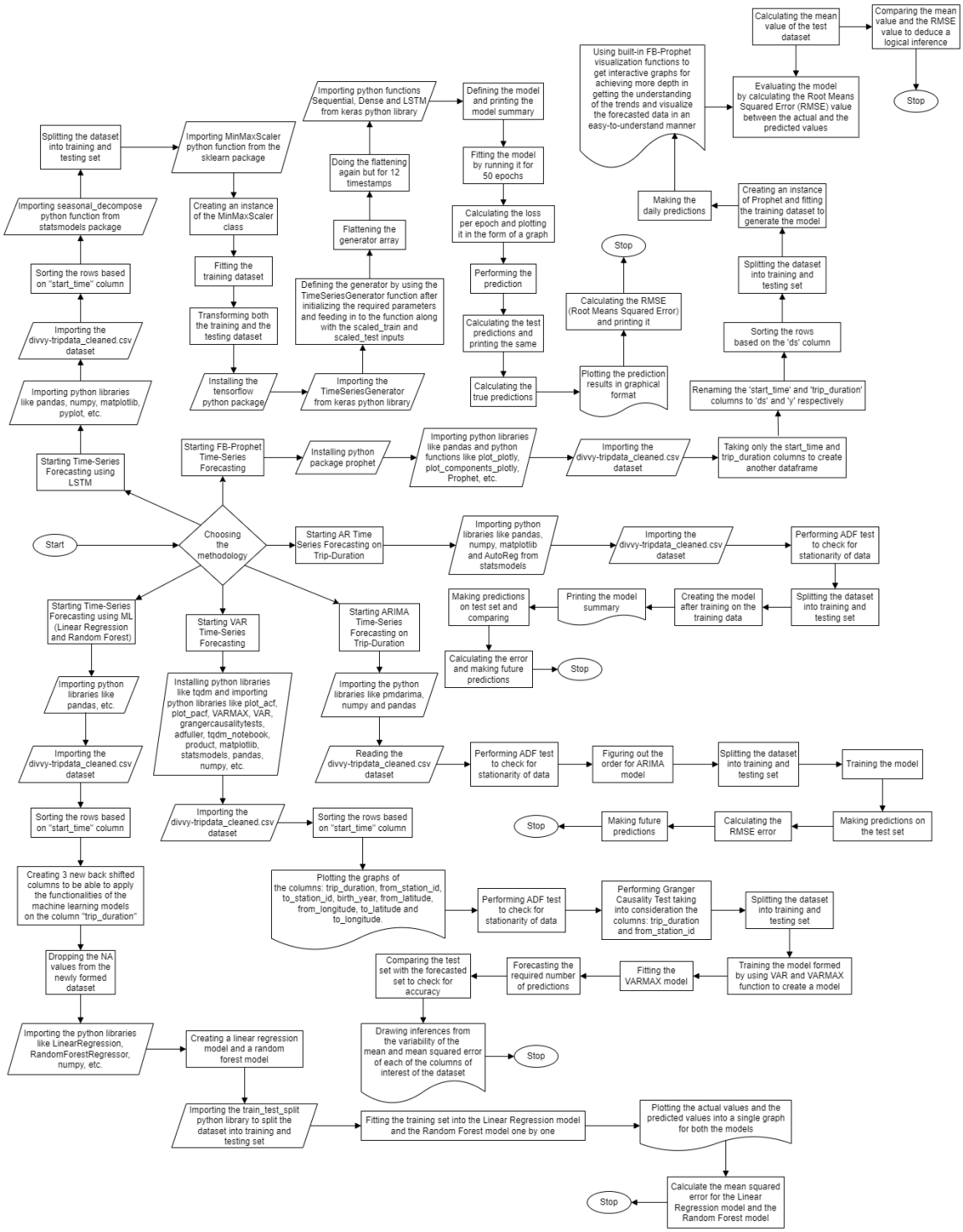


Fig. 25. Algorithm for various methodologies used for forecasting and analysis.

Visual aid to get more insights about the sampled data was also leveraged to understand the underlying trends and as a preliminary step for attempting to formulate the final marketing strategies for converting more number of customers to the subscribers of the Divvy bike-share service. Some of the visualizations were achieved with the help of

RStudio, whereas some others with basic MS Excel tips and tricks. Using RStudio and ggplot library, we obtained the graphs for the number of rides taken by each of the category of the service-users for each day of the week for the sampled data. According to the data, the customers were observed to take more number of rides than the subscribers on weekends. On the contrary, on the working week-days, the dedicated subscribers were noticed to take significantly greater number of trip rides than the customers. The average trip duration of the consumers and subscribers on various days of the week was displayed in another RStudio graph. A straight observation yielded a clear-cut inference of the customers exceeding the subscribers in terms of the average trip duration taken on each day of the week. After performing a descriptive analysis on the initial, cleaned-up, and sampled data, three datasets were generated, one for each of the following: the average trip duration and number of trips of each user type distributed across each weekday, the average trip duration and number of trips of each user type distributed across each month, and the summary of the stations. These datasets were used to generate graphs manually on MS Excel afterwards.

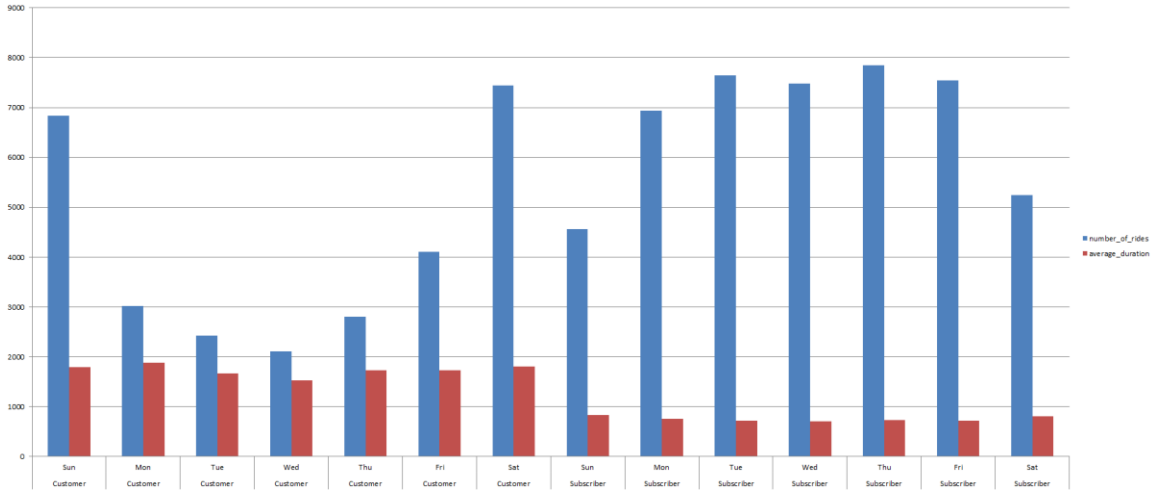


Fig. 26. Week days wise breakage of the average duration and no. of rides of customers and subscribers.

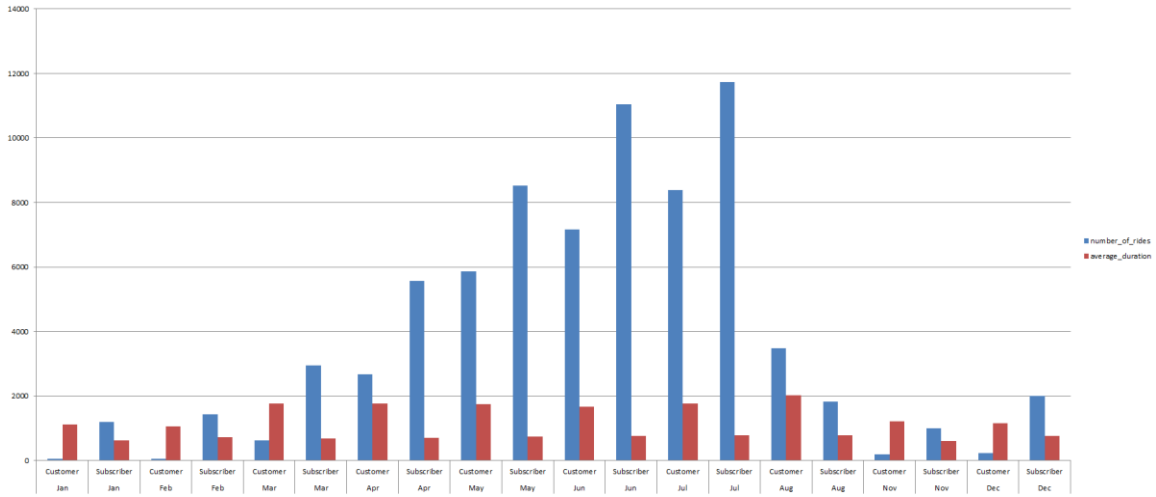


Fig. 27. Month wise breakage of the average duration and no. of rides of customers and subscribers.

Every month of the year, except for August, it was clear from the observations made from month-focused data graphs that the total number of rides taken by customers was less than the total number of rides taken by subscribers. However, the August observation was surprising because it showed that the total number of rides taken by customers were actually greater than the total number of rides taken by subscribers, even though the difference was not as large as in some extreme cases. On the other hand, the average trip duration was always higher for the customers in all the months, with the difference between the two parties remaining fairly consistent. In another effort, we tried to graphically show the most popular bike stations based upon the number of rides associated with each for both the subscribers and the customers separately. For the customers, the top 10 most popular bike docking stations came out to be Michigan Ave & Washington St, Indiana Ave & Roosevelt Rd, McClurg Ct & Illinois St, Lake Shore Dr & North Blvd, Museum Campus, Theater on the Lake, Millenium Park, Michigan Ave & Oak St, Lake Shore & Monroe St, and Streeter Dr & Illinois St in increasing order of their popularity. For the subscribers, the top 10 most popular bike docking stations came out to be Canal St & Jackson Blvd, Larrabee St & Kingsbury St, Columbus Dr & Randolph St, Clinton St & Madison St, Orleans St & Merchandise Mart Plaza, Franklin St & Arcade Pl, Dearborn St & Monroe St, Canal St & Madison St, Canal St & Adams St, and Clinton St & Washington Blvd in increasing order of their popularity.

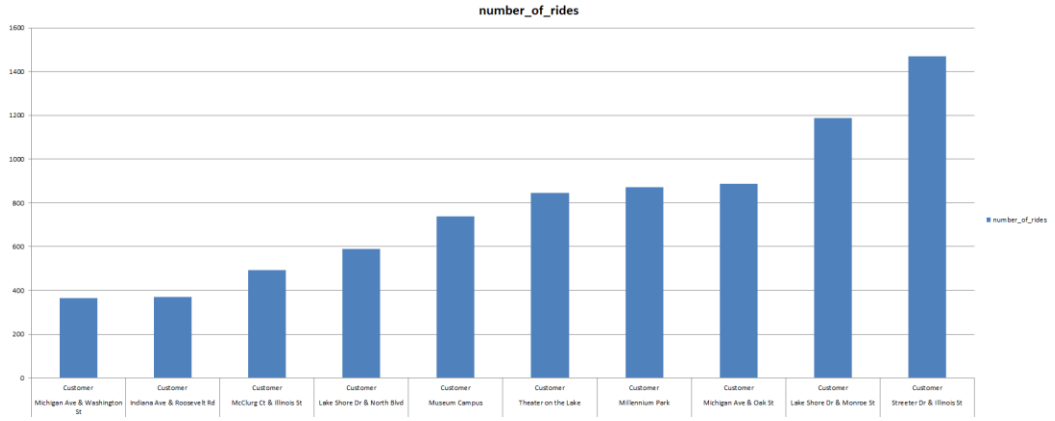


Fig. 28. Top 10 most popular to stations according to customers.

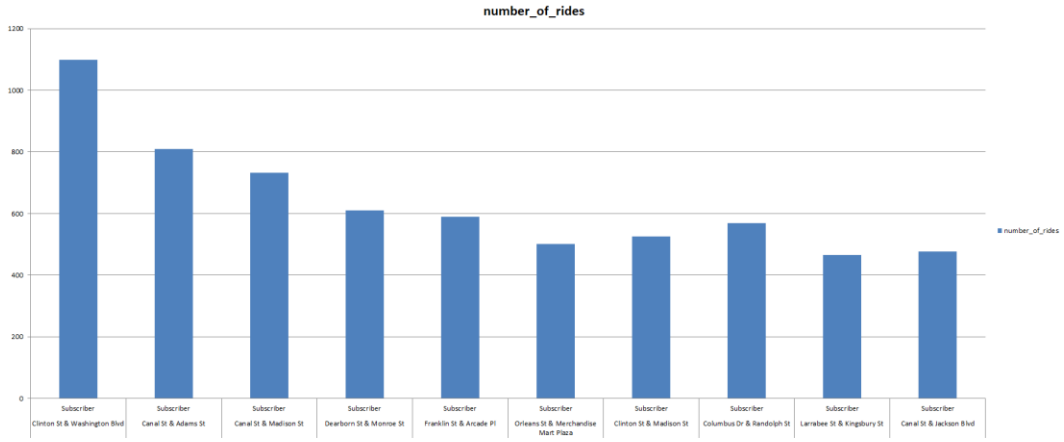


Fig. 29. Top 10 most popular to stations according to subscribers.

Darts is a very useful python library providing many useful tools to perform many data analytics operations and also time forecasting functionalities. We started by creating a new R-Script document. We updated the working directory to the script path, followed by loading the R packages like tidyverse and dplyr, aggregating the "trip_duration" for each "year-month", and writing the resultant data frame into a csv file. For doing the Darts time series forecasting, we began by installing the darts python package, importing python libraries like pandas and numpy, and importing the newly created dataset and reading the pandas dataframe. We then sorted the dataframe on the basis of "year-month/month" column, imported python libraries and functions like matplotlib, autoreload, sys, time, pyplot, datetime, reduce, functools, darts, TimeSeries, NaiveSeasonal, NaiveDrift, Prophet, ExponentialSmoothing, ARIMA, AutoARIMA, RegressionEnsembleModel, RegressionModel, Theta, FFT, mape, mse, check_seasonality, plot_acf, plot_residuals_analysis, warnings and logging, made a series from the pandas dataframe using TimeSeries function, plotted the series formed, splitted the series into training and validation sets and plotting them, created an instance of the NaiveSeasonal class (K=1) named naive_model, fitted the naive_model with the training set, plotted the actual data alongside the naive forecast (K=1), and used the plot_acf function to plot the graph, taking the lags as 9 and alpha as .05 followed by checking for any seasonality in the data.

Fig. 30. Training vs Validation graph in Darts implementation.

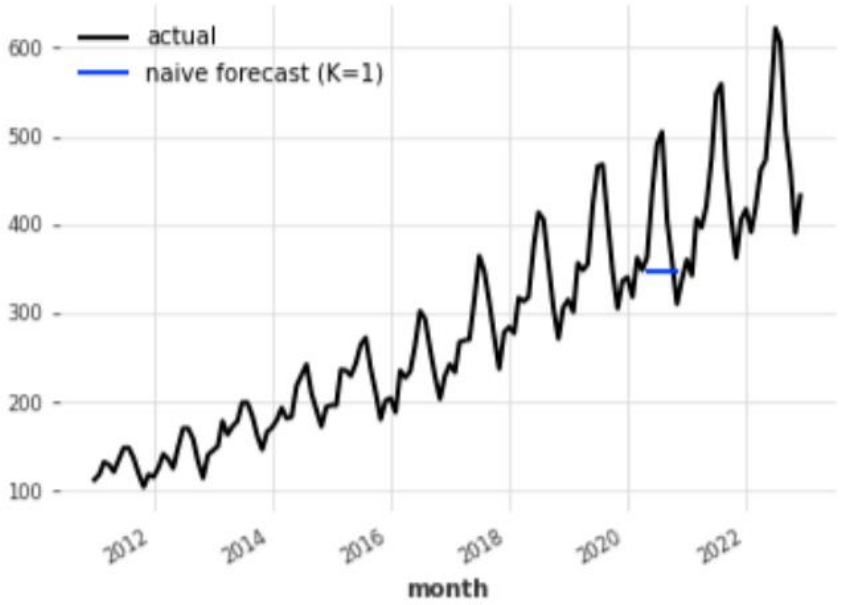


Fig. 31. The Actual data vs Naïve Forecast (K=1) data graph.

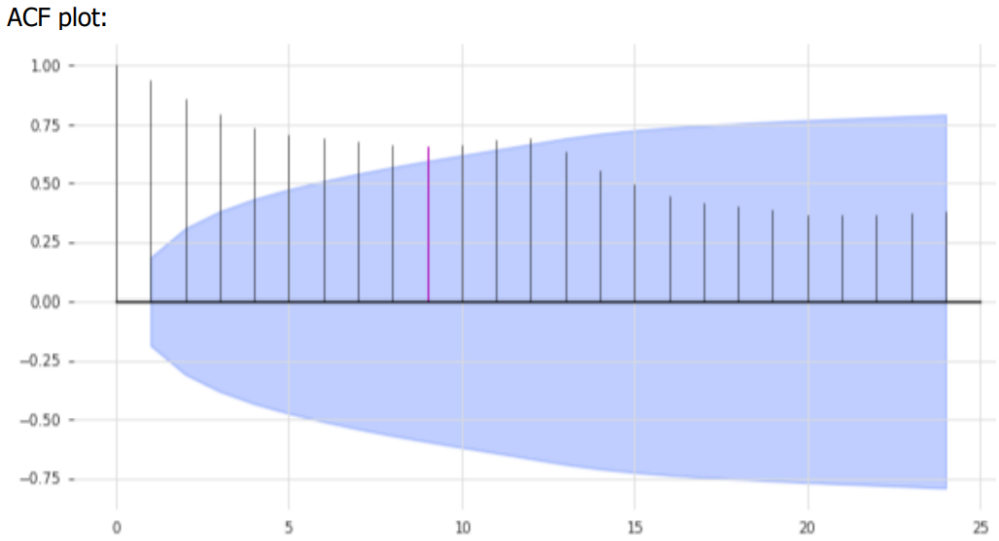


Fig. 32. Graphical representation of ACF plot.

We then created an instance of the NaiveSeasonal class (K=4) named seasonal_model, fitted the seasonal_model with the training data, created seasonal forecast for the next 120 values, plotted the actual data alongside the naive forecast (K=4), created an instance of the NaiveSeasonal class (K=13) named seasonal_model, fitted the seasonal_model with the training data, created seasonal forecast for the next 120 values, plotted the actual data alongside the naive forecast (K=13), created an instance of the NaiveSeasonal class (K=22) named seasonal_model, fitted the seasonal_model with the training data, created seasonal forecast for the next 120 values, plotted the actual data alongside the naive forecast (K=22), created an instance of the NaiveDrift class named drift_model, fitted the drift_model with the training data, created

drift forecast for the next 120 values, created combined_forecast with the drift_forecast and the seasonal_forecast eliminating the last value of the training set, plotted the combined graph of the trip_duration, combined_forecast and the drift_forecast alongside each other in the same graph, followed by calculating the mean absolute percentage error (MAPE) for the combined naive drift and calculating the MAPE for each of the models: ExponentialSmoothing, Prophet, AutoARIMA, Theta.

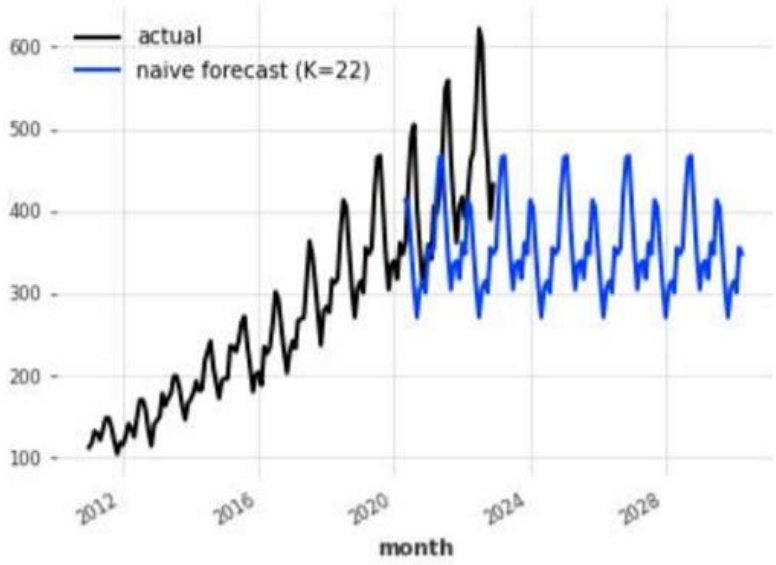


Fig. 33. The Actual data vs Naïve Forecast (K=22) data graph.

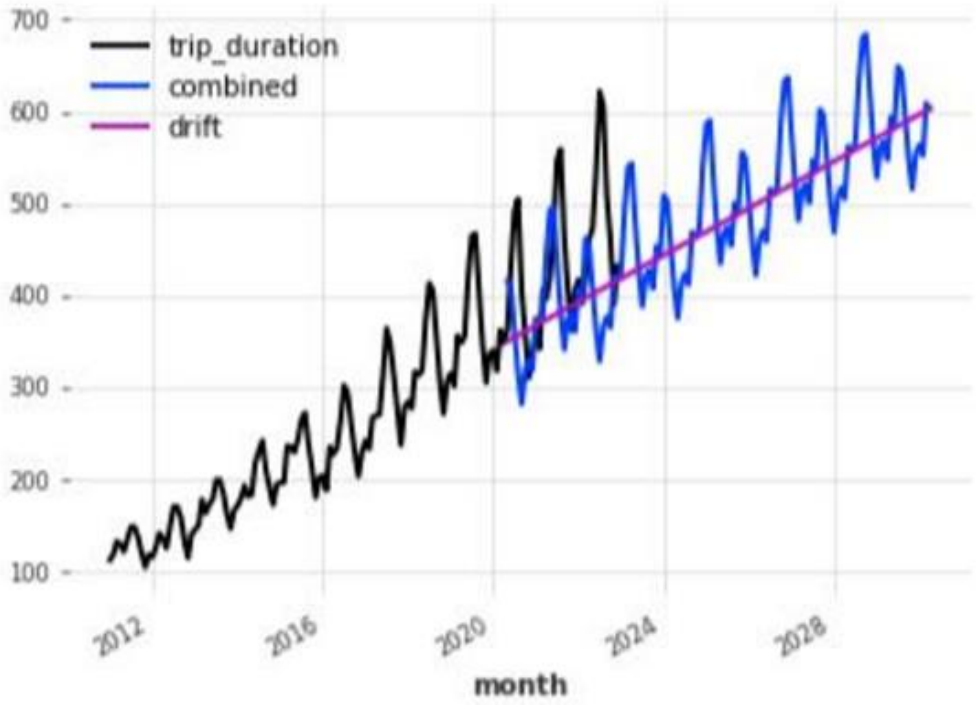


Fig. 34. The Naïve Drift combined forecast graph.

We then searched for the best theta parameter, by trying 50 different values, calculated the MAPE of the best theta model, plotted the combined graph of the best theta model prediction, calculated the Average error (MAPE) and the Median error (MAPE) over all historical forecasts and drawing the Industrial error scores (histogram) programatically, plotted the historical forecast theta data (backtest 3-months forecast (Theta)) with the original data, plotted residuals analysis over the best theta model residuals, created an instance of the ExponentialSmoothing class named model_es, formed the backtest 3-months ahead forecast (Exponential Smoothing) plot, plotted residuals analysis over the exponential smoothing model residuals, created an instance of the ExponentialSmoothing class named model_es and fitting the training data before doing the probabilistic forecast for 500 samples, followed by the plotting of the probabilistic forecast and the probabilistic forecasts for the 1-99th and 20-80th percentiles.

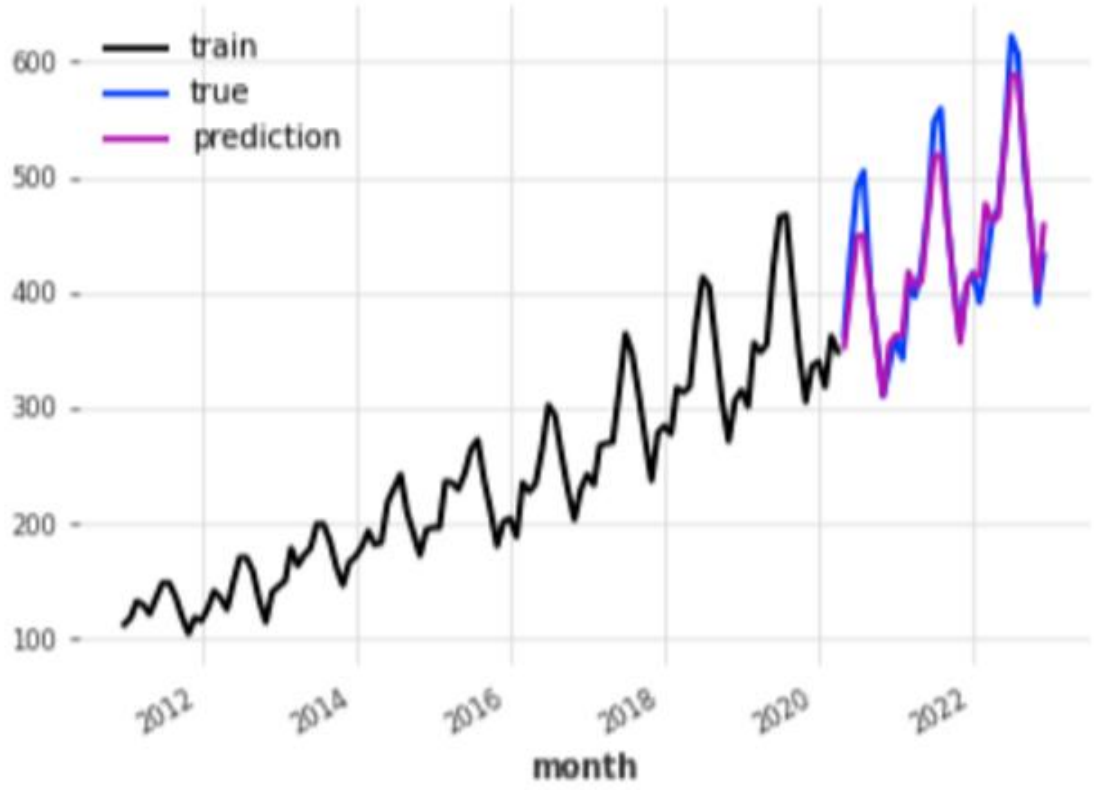


Fig. 35. The best Theta model prediction graph.

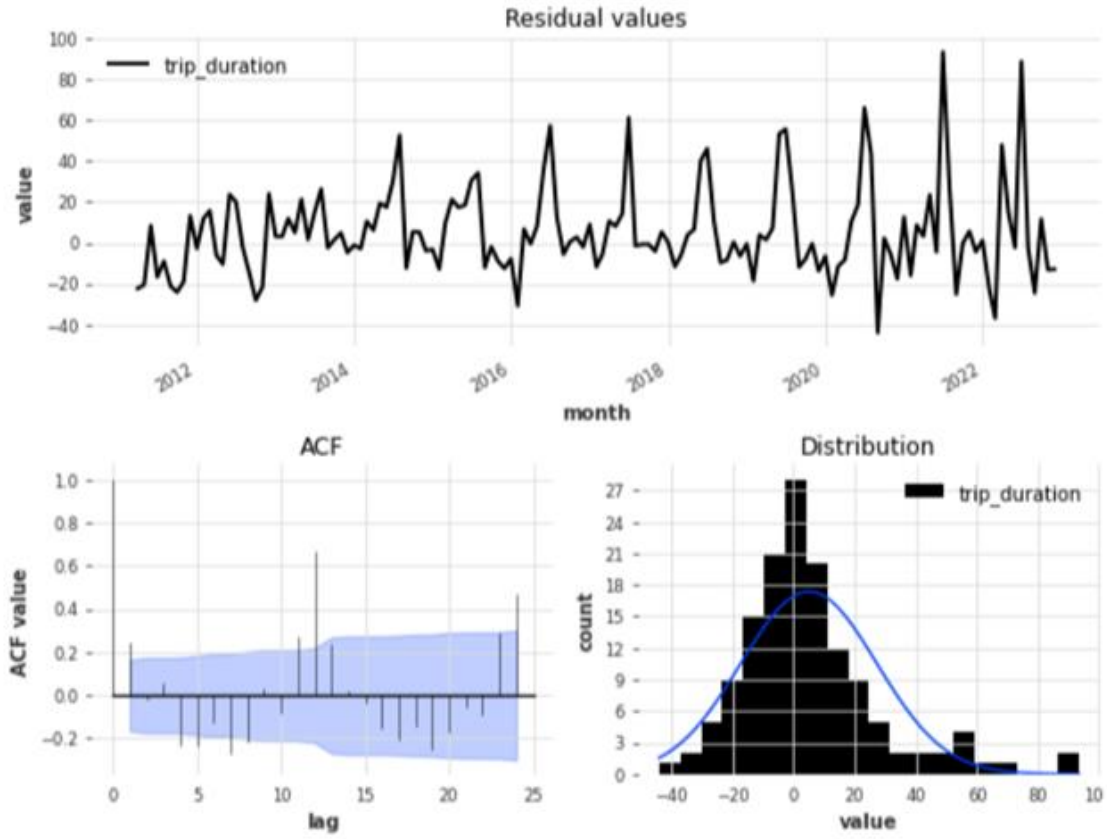


Fig. 36. The best Theta model residuals analysis graphs.

We then formed an ensemble model using forecasting models like NaiveSeasonal(6), NaiveSeasonal(12) and NaiveDrift, fitted the ensemble model with training data and predicting for 120 values, plotted the ensemble forecast graph, calculated the MAPE for the ensemble prediction, drew the ensemble forecast (historical forecast) graph, and calculated the MAPE for the historical ensemble prediction.

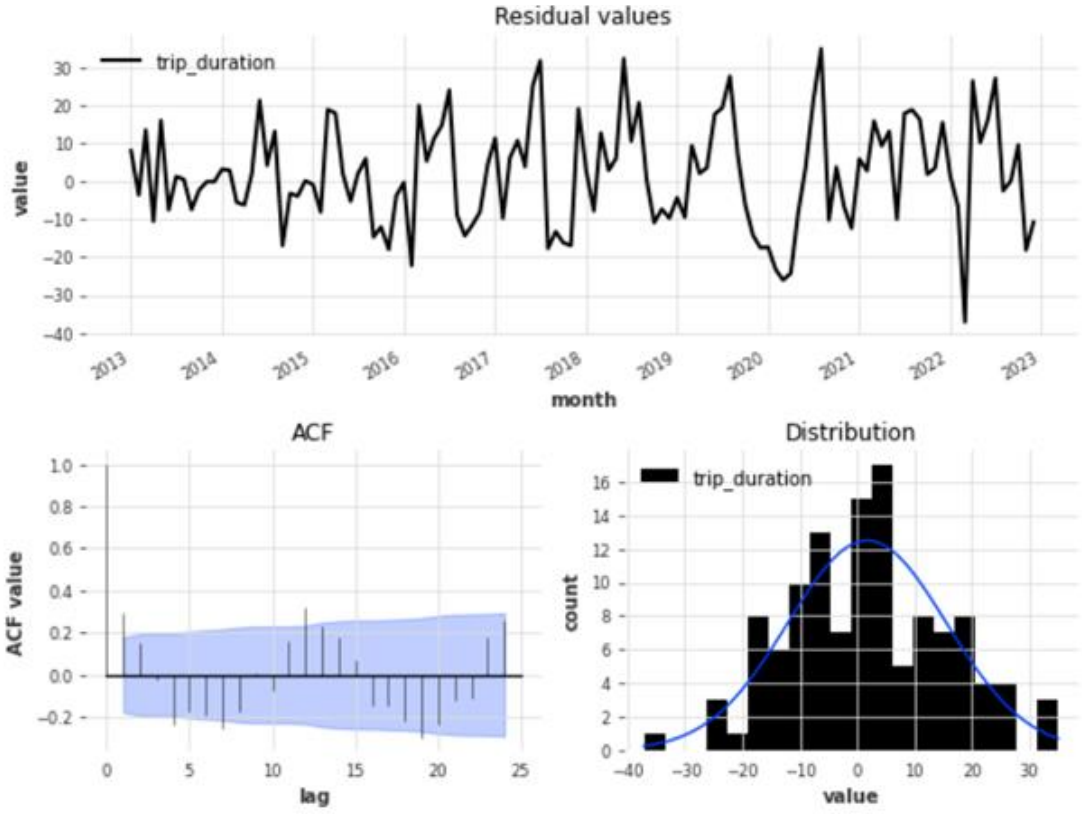


Fig. 37. The Exponential Smoothing model residuals analysis graphs.

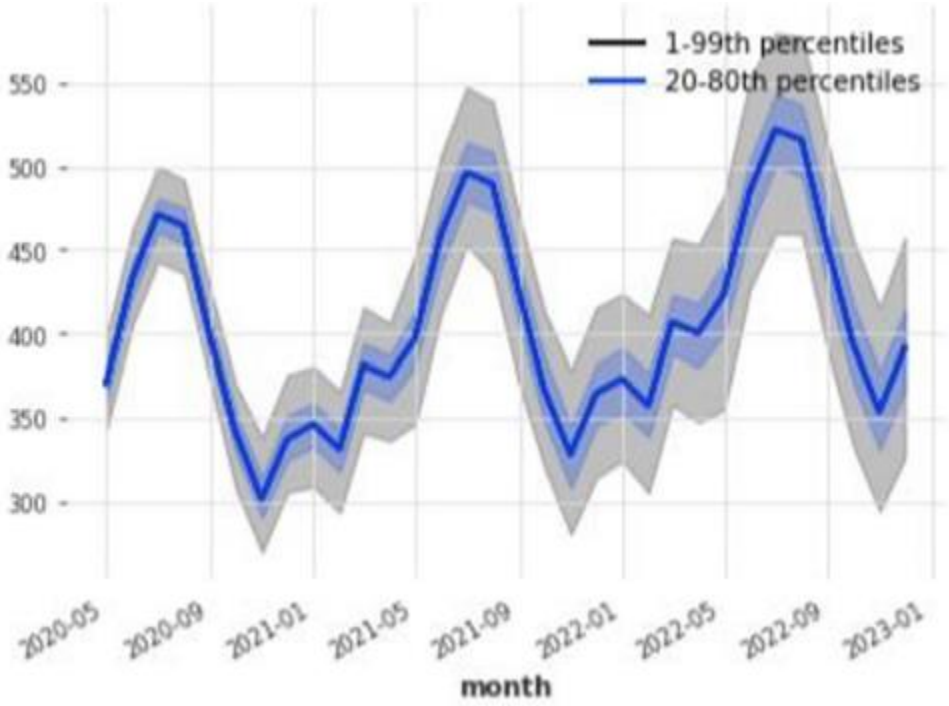


Fig. 38. Probabilistic forecast (for 1-99th and 20-80th percentiles) graph.

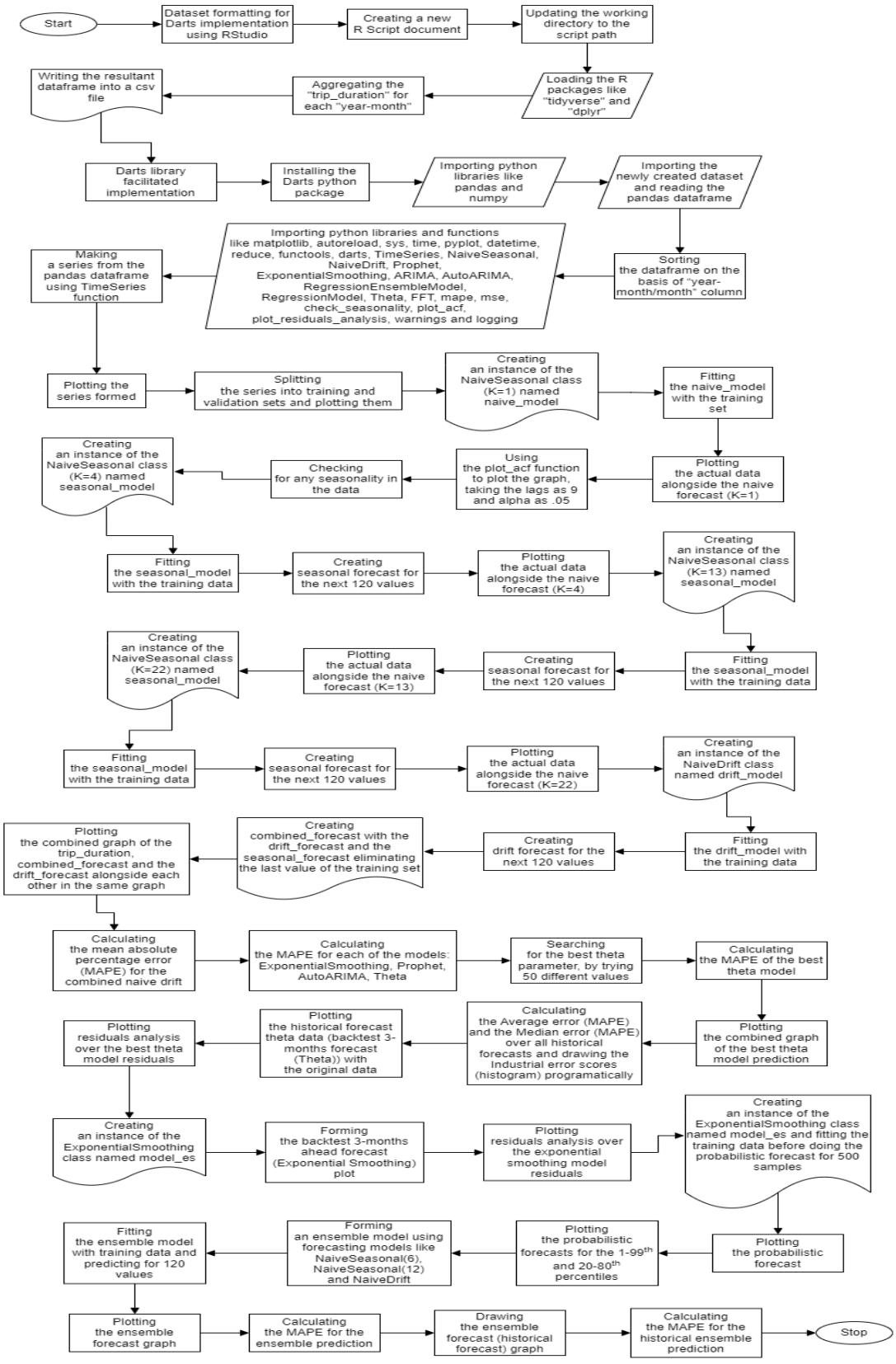


Fig. 39. Algorithm for Darts implementation.

Since XGBoost is a scalable, distributed GBDT machine learning toolkit, it has also been applied to the data for various analyses. First of all, we imported python libraries and functions such as numpy, pandas, matplotlib, pyplot, seaborn, xgboost, sklearn and mean_squared_error, followed by importing the dataset and converting it into a dataframe, sorting the dataframe on the basis of the "start_time" column, plotting the data, bifurcated the dataset into testing as well as training lots with a suitable fragmentation mark/timestamp and plotting the resultant, plotting a week of data from the starting timestamp '2014-06-30 10:51:00' extending till the ending timestamp '2014-07-07 10:51:00', creating time series features based on time series index, visualizing the feature/target relationship, creating the model using the XGBoostRegressor function and fitting it, creating a graph for realizing the feature importance of the various different parameters, forecasting on the test data and plotting a graph for the same, plotting the Truth Data and the Prediction for the specific week of data, and then finally calculating the RMSE score and the error.

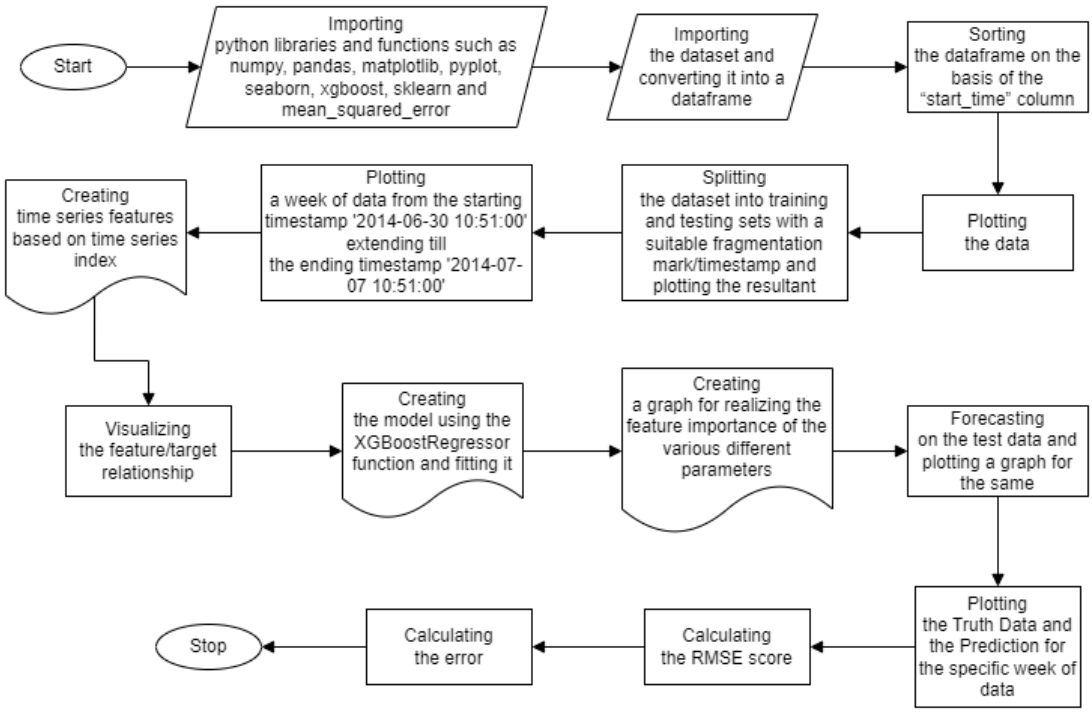


Fig. 40. Algorithm for XGBoost implementation.

Train / Test Split

```
In [5]: train=df3.loc[df3.index<'2014-06-30 10:51:00']
        test=df3.loc[df3.index>='2014-06-30 10:51:00']

        fig,ax=plt.subplots(figsize=(10,5))
        train.plot(ax=ax,label='Training Set',title='Data Train/Test Split')
        test.plot(ax=ax,label='Test Set')
        ax.axvline('2014-06-30 10:51:00',color='black',ls='--')
        plt.show()
```

Fig. 41. Code snippet for bifurcating the dataset into testing and training lots in XGBoost analysis.

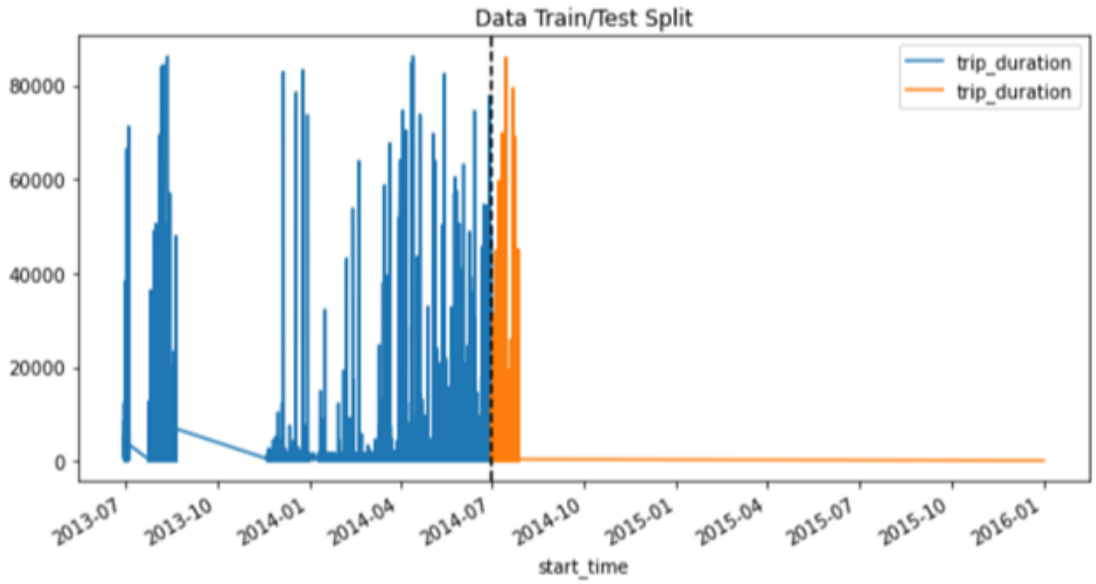


Fig. 42. Graphical representation of the train/test split in XGBoost analysis.

Feature Creation

In [7]:
```
def create_features(df3):
    """
    Create time series features based on time series index.
    """
    df3 = df3.copy()
    df3['hour'] = df3.index.hour
    df3['dayofweek'] = df3.index.dayofweek
    df3['quarter'] = df3.index.quarter
    df3['month'] = df3.index.month
    df3['year'] = df3.index.year
    df3['dayofyear'] = df3.index.dayofyear
    df3['dayofmonth'] = df3.index.day
    df3['weekofyear'] = df3.index.isocalendar().week
    return df3

df3 = create_features(df3)
```

Fig. 43. Code snippet for Feature creation in XGBoost analysis.

Create our Model

In [10]:
```
train = create_features(train)
test = create_features(test)

FEATURES = ['dayofyear', 'hour', 'dayofweek', 'quarter', 'month', 'year']
TARGET = 'trip_duration'

X_train = train[FEATURES]
y_train = train[TARGET]

X_test = test[FEATURES]
y_test = test[TARGET]
```

In [11]:
```
reg = xgb.XGBRegressor(base_score=0.5, booster='gbtree',
                       n_estimators=1000,
                       early_stopping_rounds=50,
                       objective='reg:linear',
                       max_depth=3,
                       learning_rate=0.01)
reg.fit(X_train, y_train,
        eval_set=[(X_train, y_train), (X_test, y_test)],
        verbose=100)
```

Fig. 44. Code snippet for Model creation in XGBoost analysis.

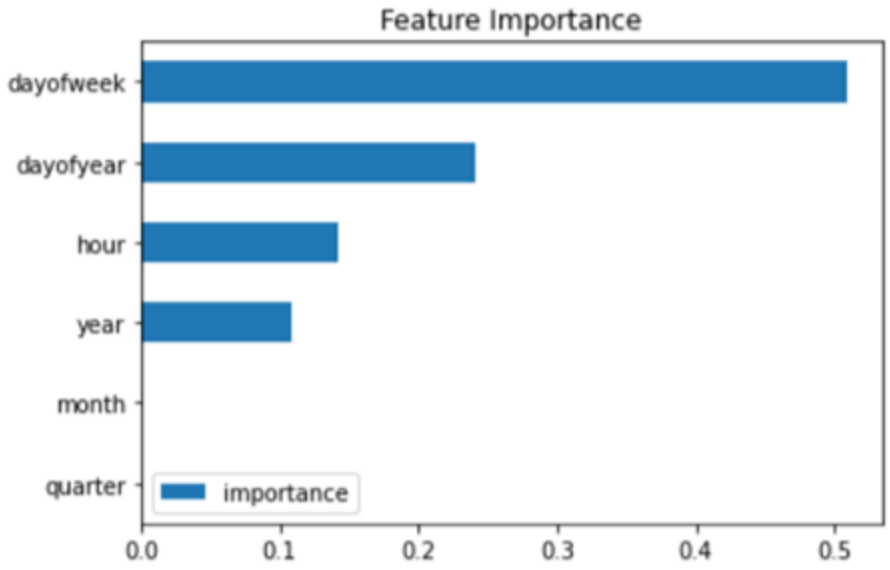


Fig. 45. Feature importance in XGBoost.

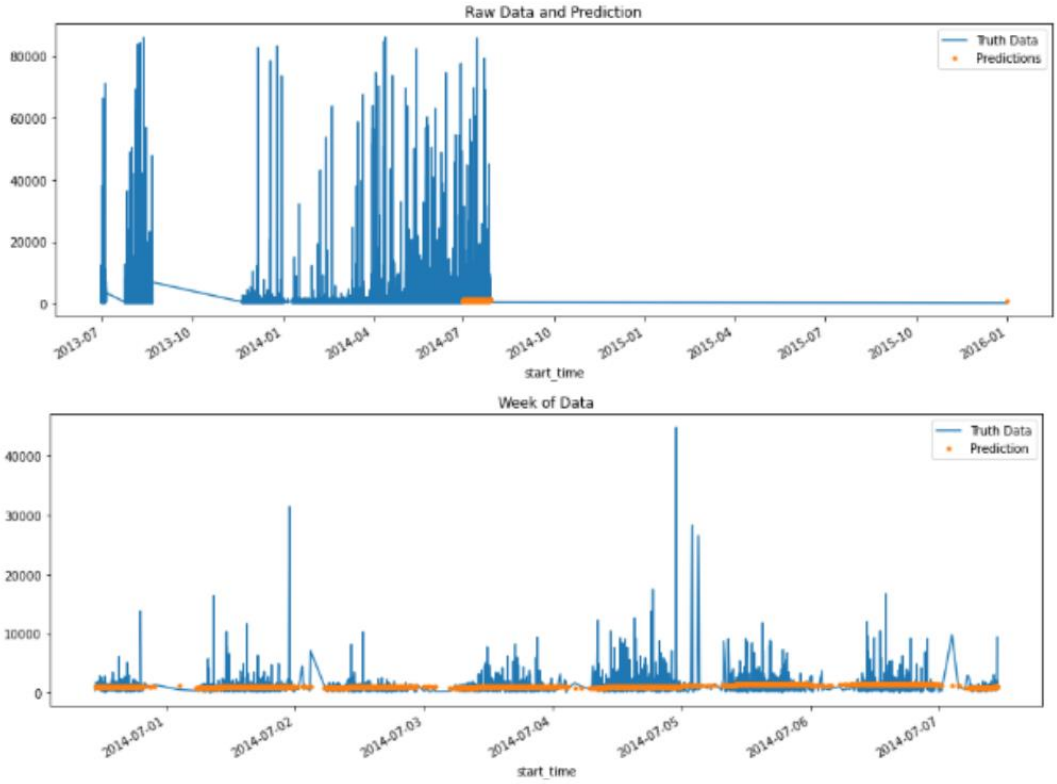


Fig. 46. Data Forecasting and Prediction in XGBoost.

Data visualizations have also been performed using Tableau. A frequently pronounced name in the world of data visualization is a rapidly flourishing and robust tool known by the name of Tableau. Its main usage is known in the business intelligence industry. If one is looking to simplify the raw data found in a certain format to an easy as

well as understandable format then Tableau is obviously the goto tool. It is such a fantabulous tool which makes it certain that the data you create could be lucidly and very very easily understood by the proficient and learned experts sitting at all the levels as well as tiers of your organization. Users who are in the non-technical domain can also use it to create customized dashboards. Tableau has some of the best features ranging from real time analysis to data blending to collaboration of data. Tableau stands out in the aspect of visualizations as it displays its results in the form of dashboards and spreadsheets and its also very fast in its working. In order to utilize Tableau we had to do some operations first. We navigated to the official Tableau website on a web browser, moved to the Tableau Public download webpage for downloading the installation file of the version of the software for public usage, clicked on the link to download the Tableau Public 2022.4 execution file (TableauPublicDesktop-64bit-2022-4-0.exe), saved the execution file to the downloads folder, double-clicked on the executive file to install the software on the system, and finally agreed to the terms and conditions and then selecting suitable preferences to finally install the software on the computer. After following all the steps, we started the Tableau Public 2022.4 application. First of all, we imported the the divvy-tripdata_cleaned.csv dataset into the workspace, then we created a new sheet for doing all the visualizations, dragged the ‘User Type’ to the Columns and the ‘Year’ onto the Rows to obtain the bar chart, followed by making other charts which included dragging ‘Gender’ to the Columns and the ‘Year’ onto the Rows, ‘Gender’ to the Columns and the ‘Day’ onto the Rows, ‘Birth Year’ to the Columns and the ‘Day’ onto the Rows, and ‘User Type’ to the Columns and the ‘Month’ onto the Rows, in order to obtain the respective charts. We also explored some other aspects of the data and familiarized ourselves with the Tableau interface.

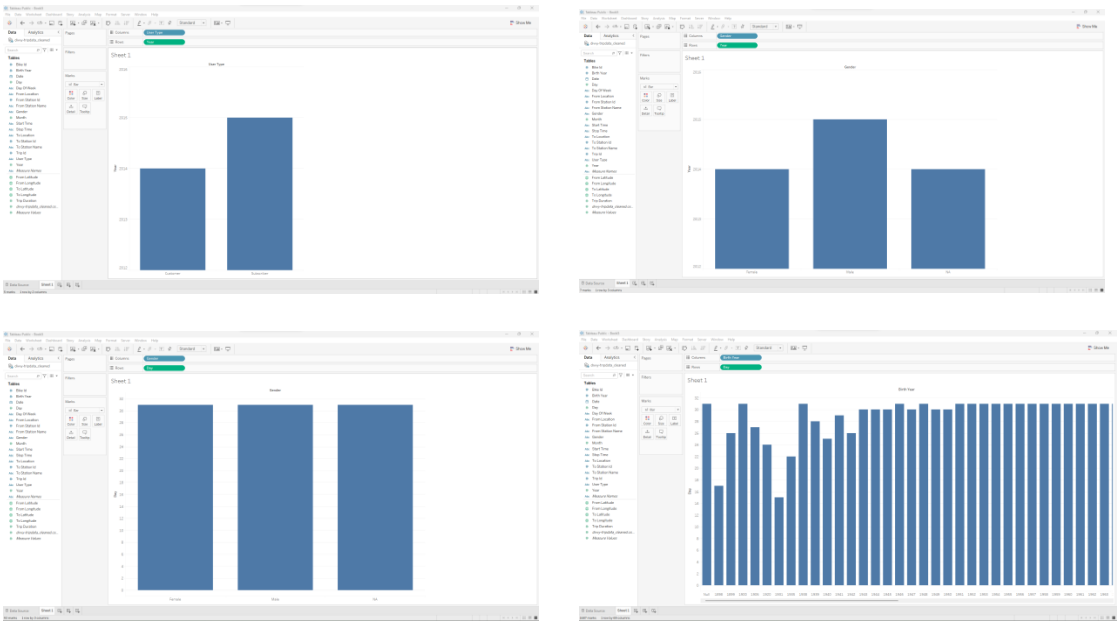


Fig. 47. Some of the data visualizations obtained using Tableau.

Exponential Smoothing was effective in our study. After importing python packages like pandas, statsmodels, etc., python functions like ExponentialSmoothing,

etc., importing the dataset and converting it into a dataframe, we decided which columns to include alongside the column acting as index, sorted the dataframe on the basis of the "start_time" column, formed the dataframe on the basis of the frequency of "start_time" column, created a model using the Exponential Smoothing function and also fitting it, specified the time series to model using the "endog" parameter, predicted the model forecast with taking steps of 100, and finally plotted the graph of the predictions.

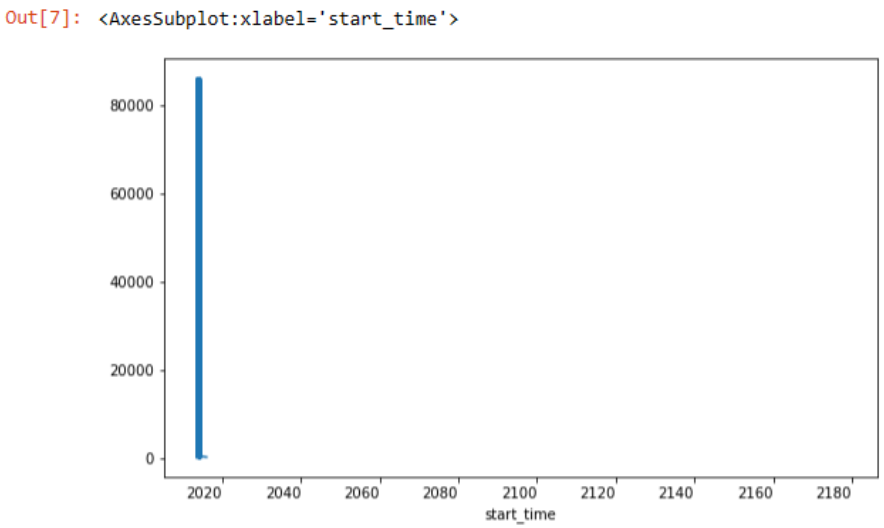


Fig. 48. Graph for prediction in Exponential Smoothing implementation.

Holt-Winters was another approach. First of all we imported the python packages like pandas, statsmodels, etc., python functions like ExponentialSmoothing, etc., imported the dataset and converted it into a dataframe, decided which columns to include alongside the column acting as index, sorted the dataframe on the basis of the "start_time" column, formed the dataframe on the basis of the frequency of "start_time" column, created a model using the ExponentialSmoothing function, specified the parameters: endog as trip_duration, trend as add, seasonal as add, and seasonal_periods as 7 and also fitted the model, predicted the model forecast with taking steps of 100, and plotted the graph of the predictions from timestamp of '2014-06-30 10:51:00' onwards.

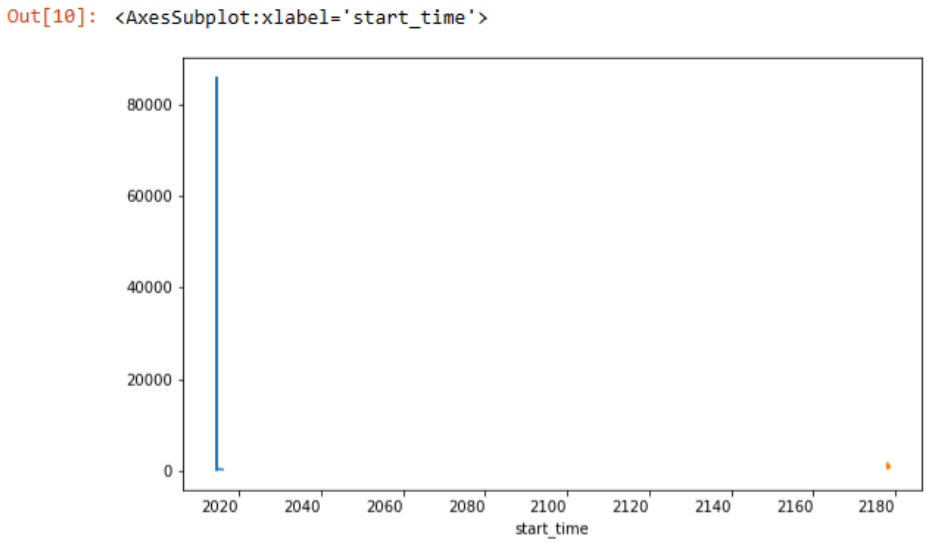


Fig. 49. Graph for prediction in Holt-Winters implementation.

The N-Beats approch for time series forecasting involved importing python libraries like pandas, numpy, datetime, matplotlib, pyplot, darts, warnings, etc., python functions like TimeSeries, etc., reading the dataset, printing the head part of the dataset, forming a series using the TimeSeries function from the dataframe, plotting the series, importing the check_seasonality function from the darts.utils.statistics package, checking for whether its daily seasonal or not and also finding out the period, checking for whether its weekly seasonal or not and also finding out the period, splitting out the series into training and testing sets, plotting the train as well as test parts, importing the NaiveSeasonal function from darts.models.forecasting.baselines package, creating a naive seasonal model by taking K as 6, fitting the naive seasonal model thus created with the train set, doing the prediction for the naive seasonal model using the items as 3, plotting the prediction results in graphical format, importing mae function from darts.metrics package, calculating the Mean Absolute Error for the naive model, printing the MAE, importing the NBEATSModel function from darts.models package, importing the Scaler class from darts.dataprocessing.transformers library, creating an instance of the scaler class and fitting it with the training set, creating the nbeats model by specifying the necessary parameters with suitable values and fitting the model, doing the prediction for N-Beats model and priting the MAE for it after calculation, importing the concatenate function from darts library, importing datetime_attribute_timeseries as dt_attr from darts.utils.timeseries_generation package, utilizing the concatenate function, creating the concatenated scaler instance and applying fit_transform, doing the prediction for it and plotting the plot for the test set, and finally calculating the MAE and printing it.

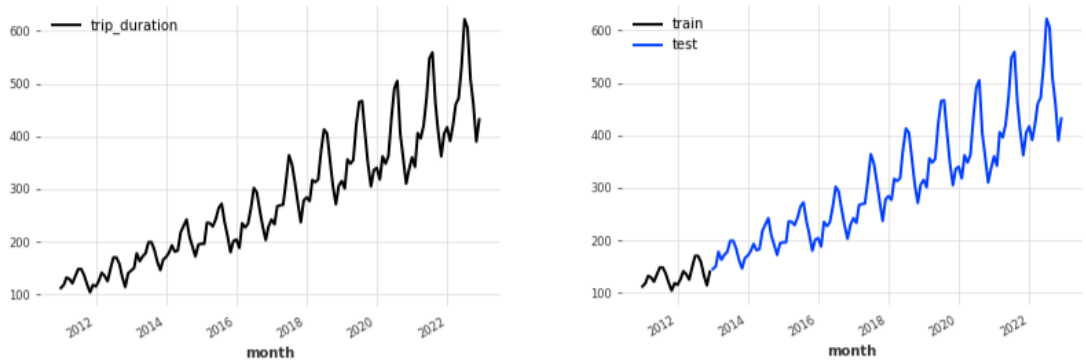


Fig. 50. Graph for representation of Actual data and Train-Test split in N-Beats approach.

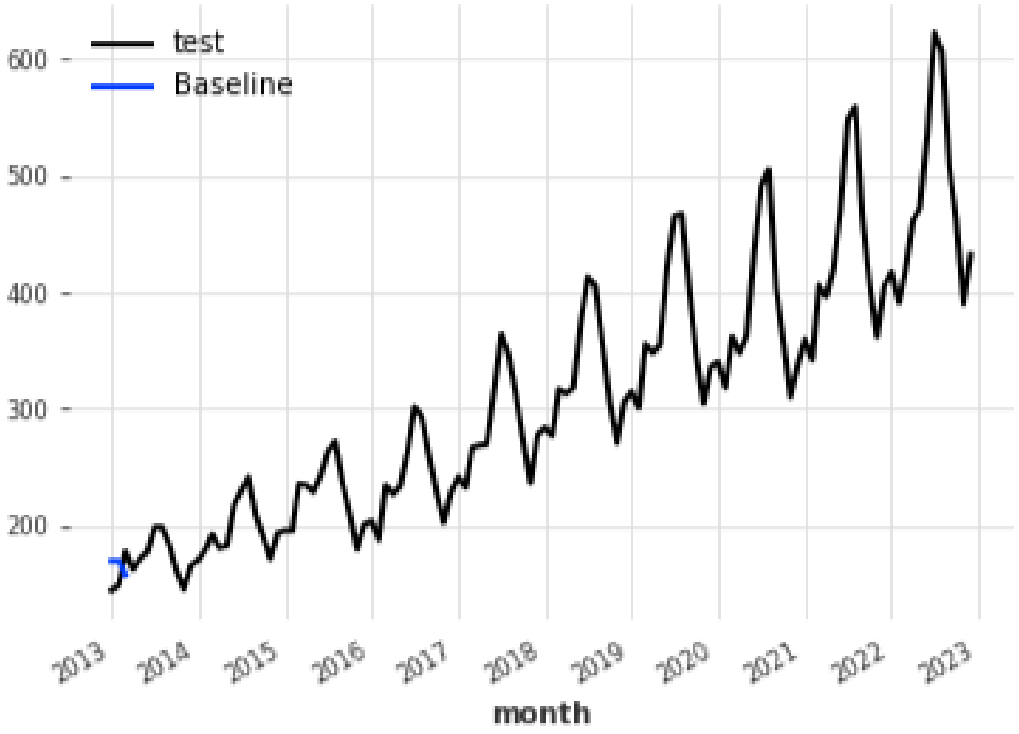


Fig. 51. Graph for representation of Test and Baseline data in NaiveSeasonal part of Darts implementation.

3.5 PARAMETER SETTING FOR ALGORITHM

The parameters used depends and varies heavily on the specific algorithm used. The first algorithm whaich was used was Vector Auto Regression (VAR), which was a time-series forecasting algorithm. It used a lot of algorithm parameters which include many types of tests as well as parameters which were actually used in determining the efficacy and even the usability of the algorithm or time-series forecasting method. The ADF Test was used to find the p-value. The p-value was of great significance since it helped us to check whether the time-series we are dealing with was actually stationary or not. The Granger Causality Test was another test used in VAR which is helpful in determining whether one

attribute determines another attribute in the dataset. In VAR Order Selection we also came across a number of parameters such as AIC, BIC, FPE, HQIC scores which had a fair amount to do.

In Auto Regressive Integrated Moving Average (ARIMA) the most important function was “pmdarima”. In this algorithms also ADF Test, p-value, AIC, BIC, FPE, HQIC scores were pronounced.

AR(1) Forecasting saw the usage of functions such as “acf”, “pacf”, “AutoReg”, etc. apart from the famous Augmented Dickey Fuller (ADF) Test and also some of the other well known parameters like p-value, AIC, BIC, FPE, HQIC scores, etc.

The advent of Machine Learning (ML) time-series forecasting algorithms namely Linear Regression and Random Forest brought a swarm of different parameters and functions into our research study.

The RNN (Recurrent Neural Network) approach, more specifically LSTM (Long Short-Term Memory) used many functions such as “Dense”, “Sequential”, “LSTM”, etc. and we got to see many new terms like loss/epoch, the Adam Optimizer, etc.

In XGBoost algorithm, we used XGBRegressor function which in turn used many parameters like “base_score”, “booster”, “n_estimators”, “_stopping_rounds”, “objective”, “max_depth”, and “learning_rate”.

Darts, Exponential Smoothing, Holt-Winters, N-Beats used internal parameters like “K”, “alpha”, “endog”, “trend”, “seasonal”, “seasonal_periods”, “max_lag”, etc.

FB Prophet used parameters such as “trend”, “yhat_lower”, “yhat_upper”, “trend_lower”, “trend_upper”, “additive_terms”, “additive_terms_lower”, “additive_terms_upper”, “daily”, “daily_lower”, “daily_upper”, “weekly”, “weekly_lower”, “weekly_upper”, “multiplicative_terms”, “multiplicative_terms_lower”, “multiplicative_terms_upper”, “yhat”, etc.

Chapter 4

Experimental Results

Based on our entire study, we were able to get many insights. Some of the major ones came from the graphical study itself. From the data sampled from between the aforementioned periods of time, its clear that the subscribers comprised of the majority of the bike services takers than the customers with the former exceeding the later by 3/5th of the total Divvy bike-share market whereas the later only has a 2/5th sharehold within the company services. The average ride duration of customers is nearly 3 times than that of subscribers. The number of rides taken by customers get doubled on weekends compared to weekdays, while the number of subscriber rides remains constant throughout the week. The subscribers average ride duration remains constant throughout the week, but the customers average ride duration fluctuates during the course of the entire week. This fluctuation might have been caused by the increase in number of customers on weekends. The average ride duration of customers does not change at the same rate as the number of rides. Still, there's a moderately positive correlation. It shows weekend customers tend to take longer rides than weekday customers. The average ride duration of customers increase and hits its peak on August but remains the same and nearly close to the peak in the months of March, April, May, June and July. The top five stations used by the customers are Lake Shore Dr & Monroe St, Michigan Ave & Oak St, Millenium Park, Theater on the Lake, and Museum Campus.

The logical conclusions which we deduced helped us to summarize our findings to a remarkable extent. Overall, customers take less number of rides but for longer durations. The customers of Divvy bike-share services mostly use bikes for recreational purposes. Unlike members who have consistent activity throughout the year, customers' use of bikes on weekends and holiday suggests they use them for recreational purposes. The customers are most active on weekends and tend to take longer rides. The customers take longest rides on the months of March, April, May, June, July and August. It peaks in the month of August.

After knowing how the customers of Divvy bike-sharing company use bikeshare differently, we were able to design recommendations and schemes to boost the sales of the company services. Designing riding packages by keeping recreational activities, weekend contests, and summer events in mind as more customers are inclined towards it can help to elevate purchase of such plans. If the customers are charged on duration basis, offered specialized discounts and coupons for regular and substantial users, this way users will be encouraged for more longer rides and thus it will result in high revenue. The designing of seasonal packages will allow flexibility and encourage customers to get membership for specific periods they want rather than paying for annual subscription. Effective and efficient promotions can be achieved by targeting customers at the busiest times and stations. The favorable days are weekends, the favorable months are June, July and August, and the most attractive stations are Lake Shore Dr & Monroe St, Michigan Ave & Oak St, and Millenium Park.

Models	RMSE	MSE	MAE	Average error (MAPE)	Median error (MAPE)
VAR	1931.66	3731328.53	-	-	-
ARIMA	2546.03	6482289.68	-	-	-
Linear Regression	1328.63	1765265.62	-	-	-
XGBoost	1919.58	3684787.38	-	-	-
Random Forest	1382.16	1910356.73	-	-	-
FBProphet	611.35	373754.17	-	-	-
AR	567.40	321947.38	-	-	-
Auto-ARIMA			-	19.09%	
LSTM	1268.50	1609083.24	-	-	-
Naive Seasonal			21.67	-	-
Naive Drift + Seasonal	-	-	-	15.67%	
Theta	-	-	-	6.66	4.74
TFT	-	-	-	-	-

Table. 1. Metrics table for all algorithms used in study.

Chapter 5

Discussion

Data analytics is key to analyzing and boosting sales of any good as well as reputable BSSs providing company. It has an upper hand when it comes to implementing market plans to target the set of customers who are the most vulnerable by suggesting them various specialized schemes and membership benefits. Hence, it is no doubt one of the most potent tools helpful in boosting the sales of a product. When it comes to companies providing bike-sharing services data analytics does have a major stand in its efficient functioning be it in the formulation of various bike-share schemes or even implementing them on a large scale. It is no joke that the directors of marketing in these companies believe that how well the company actually performs in reality in the future is leaning on the single most important factor that how many annual memberships the BSS company can gain. Hences, there shouldn't be any doubt in understanding how the two types of BSS users namely the customers and the subscribers use the services of Divvy differently. The teams involved in making all these possible can draw good and precious as well as valuable insights to appear with a new strategy which was not seen by anyone before for marketing and sales purposes of the BSSs targeting to flip the personalities of the customers or the casual users to subscribers or long-term annual members. But, as a baby step or beginners step, the executives sitting at Divvy must approve the recommendations. For doing so, nobody would have to tell them that they must be ready as well as backed up with convincing insights of the data as well as data visualizations used at the professional level.

Throughout the entire research three major questions have always guided the marketing programs of the near future:
1. How does the user characteristics between the customers (casual users) and the subscribers (long-term subscribers) actually differ?
2. Why exactly would any casual user or customer feel the urge or need to buy the company’s annual membership?
3. How can Divvy utilize the social media platforms with its social media presence to lure as well as influence the casual riders to convert to annual members?

This case study sought to understand the differences between subscriber and customer usage of the Divvy BSSs. This comparison along with other tasks was later used by the marketing department for developing strategies targeting to flip the personalities of the customers to subscribers or long-term annual members.

Chapter 6

Statistical Modelling and Analysis

After performing the ADF test in AR(1) time-series forecasting, we were able to get certain parameters. The P-Value obtained was 0.05, which implied that the data was statistically significant.

```
1. ADF :  -275.94903092250775
2. P-Value :  0.0
3. No. of Lags :  0
4. No. of Observations used for ADF Regression and Critical Value Calculation :  75999
5. Critical Values :
        1% :  -3.4304360474596383
        5% :  -2.8615780314967068
        10% :  -2.566790242857946
```

Inference: As the P-Value is less than 0.05 (statistically significant), which means the null hypothesis should be rejected. The data is stationary.

Fig. 52. Parameters obtained in ADF test of the dataset worked upon.

The Granger Causality test results in VAR implementation showed that both the dataset attributes namely “from_station_id” and “to_station_id” cause “trip_duration”.

```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=3.8872  , p=0.0487  , df_denom=75996, df_num=1
ssr based chi2 test:   chi2=3.8874  , p=0.0487  , df=1
likelihood ratio test: chi2=3.8873  , p=0.0487  , df=1
parameter F test:         F=3.8872  , p=0.0487  , df_denom=75996, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=3.6184  , p=0.0268  , df_denom=75993, df_num=2
ssr based chi2 test:   chi2=7.2373  , p=0.0268  , df=2
likelihood ratio test: chi2=7.2370  , p=0.0268  , df=2
parameter F test:         F=3.6184  , p=0.0268  , df_denom=75993, df_num=2

Granger Causality
number of lags (no zero) 3
ssr based F test:         F=6.3910  , p=0.0003  , df_denom=75990, df_num=3
ssr based chi2 test:   chi2=19.1746 , p=0.0003  , df=3
likelihood ratio test: chi2=19.1722 , p=0.0003  , df=3
parameter F test:         F=6.3910  , p=0.0003  , df_denom=75990, df_num=3

Granger Causality
number of lags (no zero) 4
ssr based F test:         F=4.9745  , p=0.0005  , df_denom=75987, df_num=4
ssr based chi2 test:   chi2=19.9004 , p=0.0005  , df=4
likelihood ratio test: chi2=19.8978 , p=0.0005  , df=4
parameter F test:         F=4.9745  , p=0.0005  , df_denom=75987, df_num=4
to station id cause trip duration?
```

```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=12.8014 , p=0.0003  , df_denom=75996, df_num=1
ssr based chi2 test:   chi2=12.8019 , p=0.0003  , df=1
likelihood ratio test: chi2=12.8009 , p=0.0003  , df=1
parameter F test:         F=12.8014 , p=0.0003  , df_denom=75996, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=11.2880 , p=0.0000  , df_denom=75993, df_num=2
ssr based chi2 test:   chi2=22.5774 , p=0.0000  , df=2
likelihood ratio test: chi2=22.5740 , p=0.0000  , df=2
parameter F test:         F=11.2880 , p=0.0000  , df_denom=75993, df_num=2

Granger Causality
number of lags (no zero) 3
ssr based F test:         F=8.2098  , p=0.0000  , df_denom=75990, df_num=3
ssr based chi2 test:   chi2=24.6315 , p=0.0000  , df=3
likelihood ratio test: chi2=24.6275 , p=0.0000  , df=3
parameter F test:         F=8.2098  , p=0.0000  , df_denom=75990, df_num=3

Granger Causality
number of lags (no zero) 4
ssr based F test:         F=6.2613  , p=0.0000  , df_denom=75987, df_num=4
ssr based chi2 test:   chi2=25.0481 , p=0.0000  , df=4
likelihood ratio test: chi2=25.0439 , p=0.0000  , df=4
parameter F test:         F=6.2613  , p=0.0000  , df_denom=75987, df_num=4
```

Fig. 53. Granger Causality test results in VAR implementation.

The gradual decrease of the loss per. Epoch in LSTM implementation showed that the efficiency of the model used to gradually go up, as the loss per. Epoch parameter gradually went down from 0.000730 to 0.000722 as the number of epochs increased from 0 to 50.

The RMSE value in most of the approaches was quite less than the MSE values, which showed that the models were good and relevant.

The graphs obtained were also complying to the business logic and they were found to support the formulation of convincing schemes and recommendations for users to buy.

Chapter 7

Conclusion and Future Work

By our study we were able to get a multi faceted view of the bike share infrastructure of the Divvy and also peep through the various parameters and statistical data of the company based in Chicago. We applied various pre-existing machine learning algorithms some of which are ubiquitous when it comes to doing time-series forecasting, besides using our own novel algorithm for achieving the feat. The various statistical figures we got by performing all the tests, the various types of visualizations we obtained by using some of the most dominant data science tools currently being used in the industry such as Tableau gave us a different view over the entire bike-share scenario in the region of our study.

Based on our endeavoring efforts and never ending hardwork to scourge the data science community to learn about the latest and the best algorithms and scientific tools, we can confidently assert that our work is one of the most comprehensive and detailed work ever done especially in this field of research. Its important to point out that in some of the time-series forecasting algorithms we were able to achieve considerable accuracy and the predictions were also upto the mark.

Nevertheless, its noteworthy that there is always a scope of improvement in any research study. We focussed on a single case study confined to a specific region of the globe but more generic ones can be done with proper planning and severe ambition in mind. Some of the pre-existing algorithms were used didn't gave us satisfactory results and metrics, but still we included them in our study for future researchers to dive into and understand whether the problem actually was, be in the wrong way of implementation or even for the shortcomings of the specific algorithm itself to map the scenario. It would also be unjust to not speak of our novel algorithm, which despite performing quite well did show a lot of vices. It is upto the future researchers to pin-point those weaknesses and improve it and make it an even bigger grand success. Although we tried to cover the challenge of bike-rebalancing in this research work, we were not successful and our derived methodologies were not upto the mark and thus we decided to scrap that dimention of approach towards solving the problem but its not impossible and has great scope of implementation in the future studies.

Appendices

Appendix 1: Data cleaning, descriptive statistics and data visualization in R

SAMPLE CODE :-

```
# update working directory to this script path
library(rstudioapi)
script_dir = rstudioapi::getActiveDocumentContext()$path
setwd(dirname(script_dir))
getwd()
```

```
# load the csv file for analysis
df <- read_csv("divvy_trips.csv")
str(df)

# taking a sample from the data for simplicity
sample_df <- sample_n(df,76000,replace=F)

# writing the sample into a sample dataset
write_csv(sample_df,"sample_dataset.csv")
str(sample_df)

# loading the sample dataset
df <- read_csv("sample_dataset.csv")

# exploring various attributes of sample
colnames(df)
nrow(df)
dim(df)
head(df)
str(df)
summary(df)
```

```r
# viewing the sample dataset
View(df)
```

```r
# checking if we have the required column inclusions
head(df)

# cleaning column names and removing duplicates
df <- df %>%
  clean_names() %>%
  unique()

# change the column name of day_of_week_weekdays_df_date to day_of_week
colnames(df)[23] <- "day_of_week"

# exporting cleaned df to new csv
write_csv(df,'divvy-tripdata_cleaned.csv')

# loading the cleaned dataset into df
df <- read_csv('divvy-tripdata_cleaned.csv')

# viewing the cleaned dataset
View(df)

# descriptive data analysis

# descriptive analysis on trip_duration
summary(df$trip_duration)
# comparing subscribers and customers
aggregate(df$trip_duration ~ df$user_type, FUN=mean)
aggregate(df$trip_duration ~ df$user_type, FUN=median)
aggregate(df$trip_duration ~ df$user_type, FUN=max)
```

```
aggregate(df$trip_duration ~ df$user_type, FUN=min)
```

Appendix 2: AR Time Series Forecasting

SAMPLE CODE :-

```
# import csv dataset
df=pd.read_csv('divvy-tripdata_cleaned.csv',index_col=1,parse_dates=True)

# make into dataframe
df2=pd.DataFrame(df)

# which columns to include alongside column acting as index
cols=[3]
df2=df2[df2.columns[cols]]
X=df2.values

# printing stuffs
print('Shape of data \t',df2.shape)
print('Original Dataset:\n',df2.head())
print('After extracting only trip duration:\n',X)
```

```
# performing "Augmented Dickey Fuller" test to check for stationarity of data

from statsmodels.tsa.stattools import adfuller

dftest=adfuller(df2['trip_duration'],autolag='AIC')

print("1. ADF : ",dftest[0])
print("2. P-Value : ",dftest[1])
print("3. No. of Lags : ",dftest[2])
print("4. No. of Observations used for ADF Regression and Critical Value
Calculation : ",dftest[3])
```

```
print("5. Critical Values :")
for key,val in dftest[4].items():
    print("\t",key, ": ",val)
```

```
# splitting the dataset into training and testing sets
train=X[:len(X)-7]
test=X[len(X)-7:]
```

```
# creating the model after training on the training data
model=AutoReg(train,lags=10).fit()
```

```
# printing the model summary
print(model.summary())
```

```
# making predictions on test set and compare
pred=model.predict(start=len(train),end=len(X)-1,dynamic=False)
```

```
from matplotlib import pyplot
pyplot.plot(pred)
pyplot.plot(test,color='red')
print(pred)
```

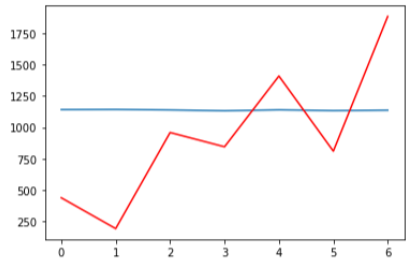
```
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse=sqrt(mean_squared_error(test,pred))
```

```
# making future predictions
pred_future=model.predict(start=len(X)+1,end=len(X)+7,dynamic=False)
print("The future prediction for the next timeframe")
print(pred_future)
print('Number of Predictions Made: \t',len(pred_future))
```

SAMPLE OUTPUT :-

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                75993
Model:                     AutoReg(10)   Log Likelihood             -695864.992
Method:              Conditional MLE   S.D. of innovations           2296.699
Date:               Mon, 05 Dec 2022   AIC                        1391753.983
Time:                       12:19:39   BIC                        1391864.843
Sample:                           10   HQIC                       1391788.047
                               75993
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const       1145.1613     15.513     73.821      0.000    1114.757    1175.566
y.L1          -0.0010      0.004     -0.266      0.790      -0.008       0.006
y.L2          -0.0028      0.004     -0.773      0.440      -0.010       0.004
y.L3           0.0013      0.004      0.364      0.716      -0.006       0.008
y.L4          -0.0011      0.004     -0.290      0.772      -0.008       0.006
y.L5           0.0041      0.004      1.143      0.253      -0.003       0.011
y.L6          -0.0063      0.004     -1.725      0.084      -0.013       0.001
y.L7          -0.0010      0.004     -0.270      0.787      -0.008       0.006
y.L8          -0.0010      0.004     -0.282      0.778      -0.008       0.006
y.L9          -0.0013      0.004     -0.365      0.715      -0.008       0.006
y.L10          0.0011      0.004      0.299      0.765      -0.006       0.008
                                    Roots
==============================================================================
                  Real          Imaginary           Modulus         Frequency
------------------------------------------------------------------------------
AR.1           -1.9863           -0.0000j            1.9863           -0.5000
AR.2           -1.4920           -1.0863j            1.8456           -0.3998
AR.3           -1.4920           +1.0863j            1.8456            0.3998
AR.4           -0.4857           -1.8841j            1.9457           -0.2902
AR.5           -0.4857           +1.8841j            1.9457            0.2902
AR.6            0.6506           -1.8350j            1.9469           -0.1958
AR.7            0.6506           +1.8350j            1.9469            0.1958
AR.8            1.7251           -0.9787j            1.9834           -0.0821
AR.9            1.7251           +0.9787j            1.9834            0.0821
AR.10           2.4109           -0.0000j            2.4109           -0.0000
------------------------------------------------------------------------------
```

```
[1140.88097595 1142.15828814 1138.04795025 1131.48955233 1138.94974506
 1132.76496985 1136.22582997]
```




```
The future prediction for the next timeframe
[1135.34267101 1136.5921055  1136.21029705 1136.2633281  1136.25222976
 1136.22392778 1136.25229371]
Number of Predictions Made:      7
```

Appendix 3: ML Approach

SAMPLE CODE :-

```
# creating data shifts
df3['trip_duration_LastShift']=df3['trip_duration'].shift(+1)
df3['trip_duration_2LastShift']=df3['trip_duration'].shift(+2)
df3['trip_duration_3LastShift']=df3['trip_duration'].shift(+3)
df3
```

```
# for linear regression
```

```
from sklearn.linear_model import LinearRegression
lin_model=LinearRegression()
```

```
# for random forest
from sklearn.ensemble import RandomForestRegressor
model=RandomForestRegressor(n_estimators=100,max_features=3,random_state=1)
```

```
import numpy as np
x1,x2,x3,y=df3['trip_duration_LastShift'],df3['trip_duration_2LastShift'],df3[
'trip_duration_3LastShift'],df3['trip_duration']
x1,x2,x3,y=np.array(x1),np.array(x2),np.array(x3),np.array(y)
x1,x2,x3,y=x1.reshape(-1,1),x2.reshape(-1,1),x3.reshape(-1,1),y.reshape(-1,1)
final_x=np.concatenate((x1,x2,x3),axis=1)
print(final_x)
```

```
# splitting the data between training and testing sets
from sklearn.model_selection import train_test_split
#    X_train,X_test,y_train,y_test=final_x[:-100],final_x[-100:],y2[-100:],y2[-
100:]
X_train,X_test,y_train,y_test=train_test_split(final_x,y,test_size=0.3)
```

```
# model fitting
model.fit(X_train,y_train)
lin_model.fit(X_train,y_train)
```

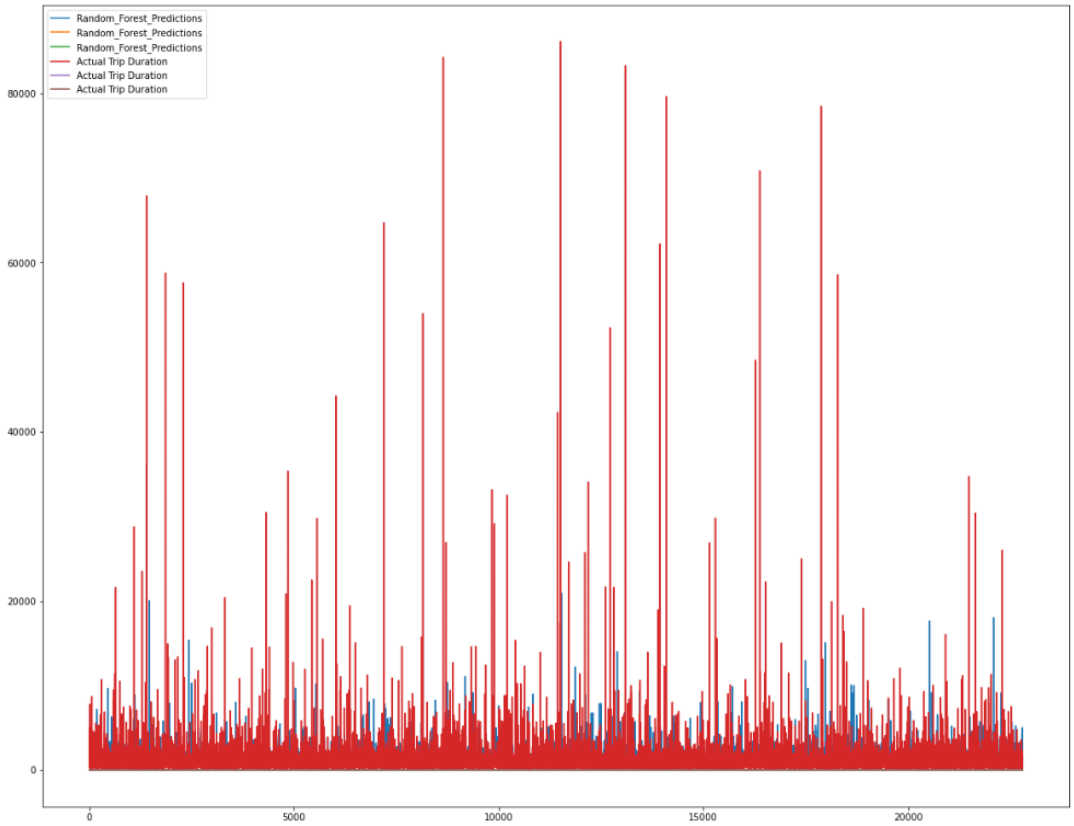
```
# random forest predictions
pred=model.predict(X_test)
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"]=(20,16)
plt.plot(pred,label='Random_Forest_Predictions')
plt.plot(y_test,label='Actual Trip Duration')
plt.legend(loc="upper left")
plt.show()
```

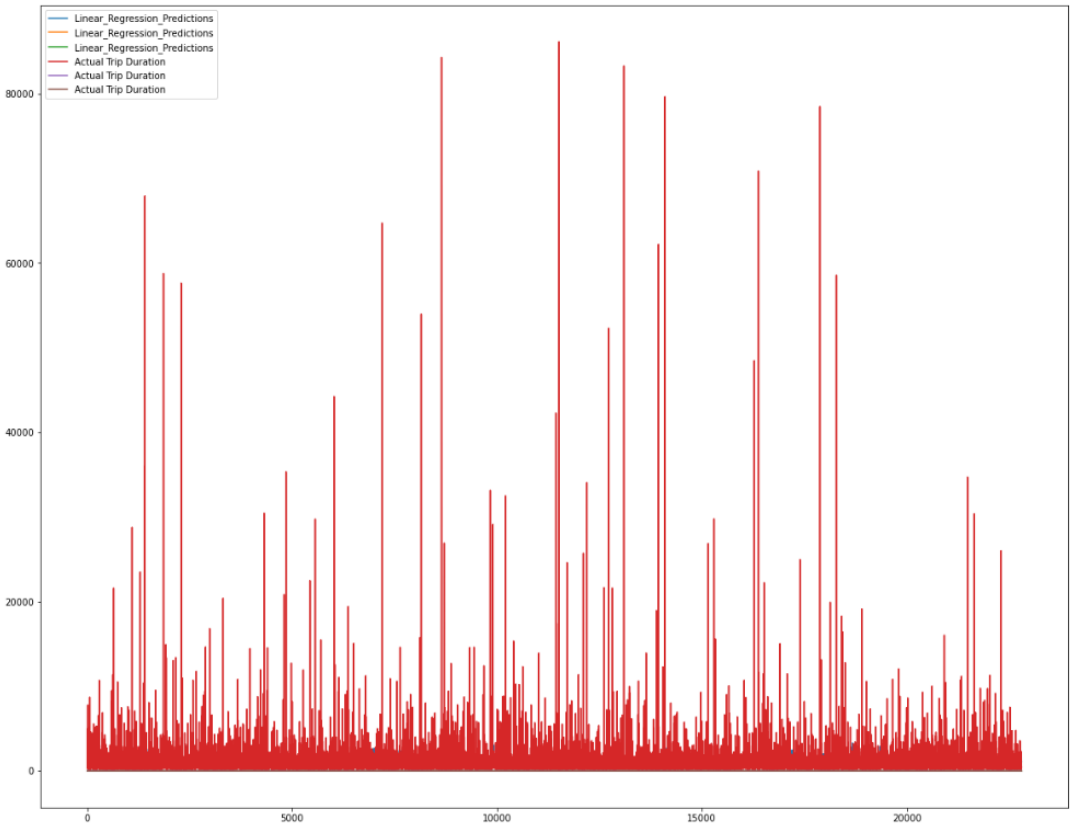
```
# linear regression predictions
lin_pred=lin_model.predict(X_test)
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"]=(20,16)
plt.plot(lin_pred,label="Linear_Regression_Predictions")
plt.plot(y_test,label="Actual Trip Duration")
plt.legend(loc="upper left")
plt.show()
```

```
# calculating the error for both the models
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse_rf=sqrt(mean_squared_error(pred,y_test))
rmse_lr=sqrt(mean_squared_error(lin_pred,y_test))
```

```
print('Mean Squared Error for Random Forest Model is:',rmse_rf)
print('Mean Squared Error for Linear Regression Model is:',rmse_lr)
```

SAMPLE OUTPUT :-


Random_Forest_Predictions
Random_Forest_Predictions
Random_Forest_Predictions
Actual Trip Duration
Actual Trip Duration
Actual Trip Duration
80000
60000
40000
20000
0
0
5000
10000
15000
20000


Linear_Regression_Predictions
Linear_Regression_Predictions
Linear_Regression_Predictions
Actual Trip Duration
Actual Trip Duration
Actual Trip Duration
80000
60000
40000
20000
0
0
5000
10000
15000
20000

```
Mean Squared Error for Random Forest Model is: 1382.156551144884
Mean Squared Error for Linear Regression Model is: 1328.632988068834
```

Appendix 4: FB Prophet

SAMPLE CODE :-

```
import pandas as pd
from prophet.plot import plot_plotly,plot_components_plotly
from prophet import Prophet
```

```
# import csv dataset
df=pd.read_csv('divvy-tripdata_cleaned.csv')


# drop the na values
df.dropna(inplace=True)
df.reset_index(drop=True,inplace=True)
```

```
# changing column names for fb prophet
df.columns=['ds','y']
```

```
df['ds']=pd.to_datetime(df['ds'])
df=df.sort_values("ds")
df.tail()
```

```
# splitting the data between training and testing sets
train=df.iloc[:len(df)-472]
test=df.iloc[len(df)-472:]
```

```
# starting to make the predictions
m=Prophet()
m.fit(train)


# making daily predictions
future=m.make_future_dataframe(periods=207,freq="D")
forecast=m.predict(future)
```

```
forecast[['ds','yhat','yhat_lower','yhat_upper']].tail()
```

```
# visualizations using built-in fb prophet tools
plot_plotly(m,forecast)
plot_components_plotly(m,forecast)
```

```
# evaluating the model
from statsmodels.tools.eval_measures import rmse

predictions=forecast.iloc[-472:]['yhat']

print("Root Mean Squared Error between actual and predicted values: ",rmse(predictions,test['y']))
print("Mean Value of Test Dataset: ",test['y'].mean())
```

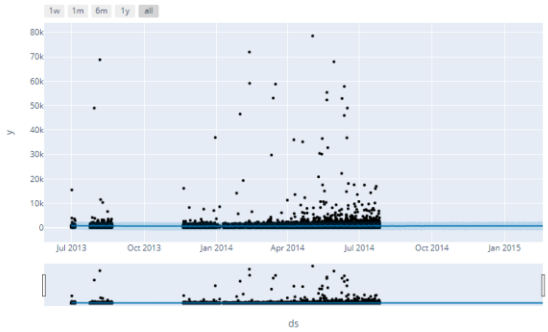
SAMPLE OUTPUT :-

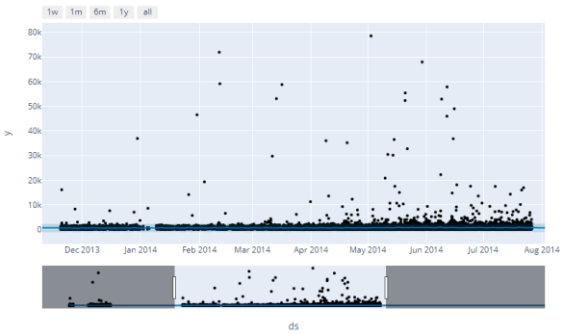
Out[13]:

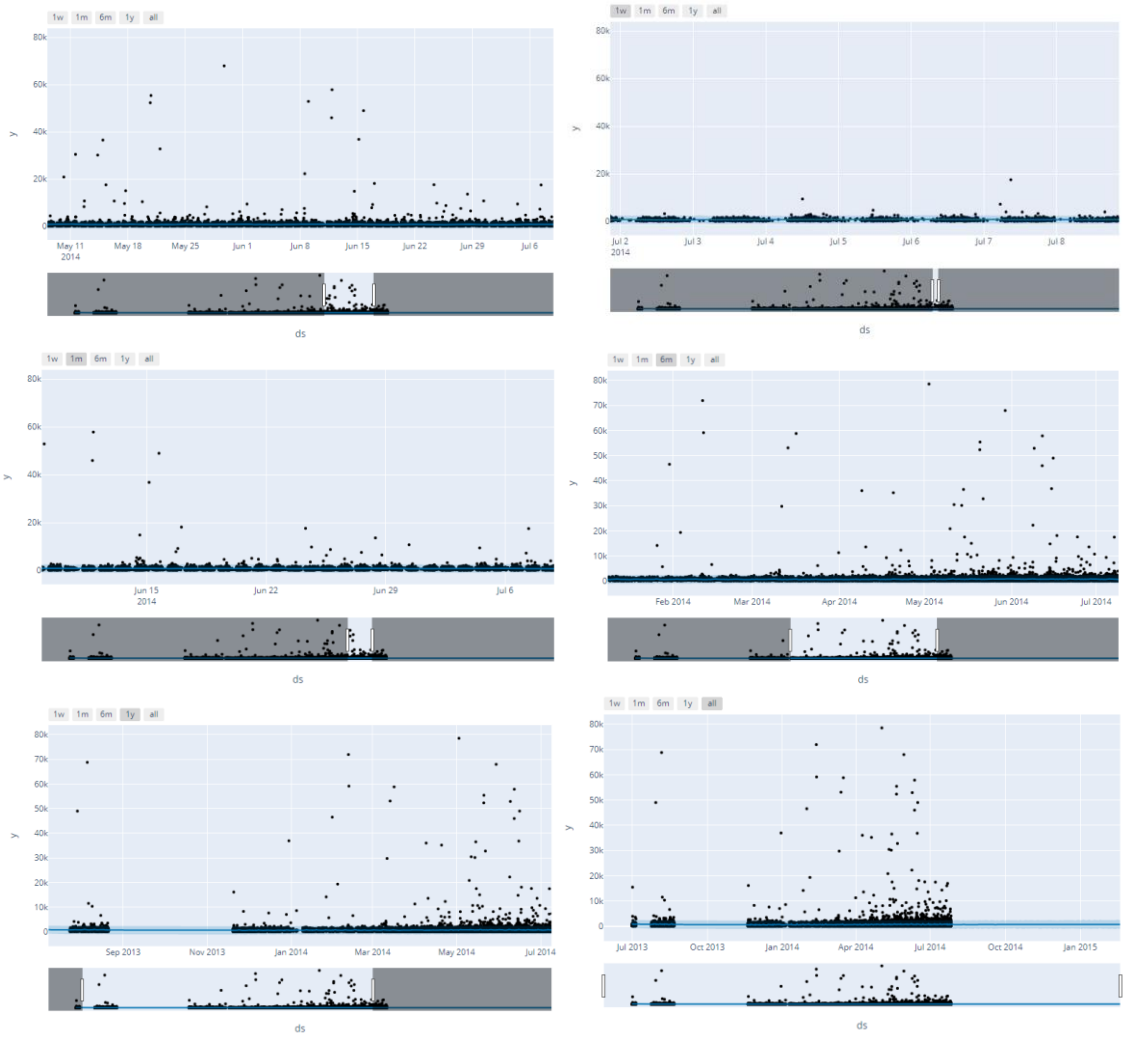
	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	daily	daily_lower	daily_upper	weekly	weekly_lower	weekly_upper	multiplicative_terms	multiplicative_terms_lower	multiplicative_terms_upper	yhat
39588	2015-02-14 13:10:00	849.717445	-877.723277	2561.838850	387.762143	1322.063831	37.073657	37.073657	37.073657	-33.361508	-33.361508	-33.361508	70.435166	70.435166	70.435166	0.0	0.0	0.0	886.791103
39589	2015-02-15 13:10:00	850.094168	-759.310288	2643.808433	382.885594	1324.660492	57.478614	57.478614	57.478614	-33.361508	-33.361508	-33.361508	90.840122	90.840122	90.840122	0.0	0.0	0.0	907.572781
39590	2015-02-16 13:10:00	850.470890	-867.498784	2609.175552	377.898944	1326.560116	-56.261059	-56.261059	-56.261059	-33.361508	-33.361508	-33.361508	-22.899551	-22.899551	-22.899551	0.0	0.0	0.0	794.209831
39591	2015-02-17 13:10:00	850.847612	-1032.077698	2586.746672	372.679401	1329.850814	-52.470053	-52.470053	-52.470053	-33.361508	-33.361508	-33.361508	-19.108545	-19.108545	-19.108545	0.0	0.0	0.0	798.377558
39592	2015-02-18 13:10:00	851.224334	-817.890385	2599.409994	369.755401	1333.514931	-77.659031	-77.659031	-77.659031	-33.361508	-33.361508	-33.361508	-44.297522	-44.297522	-44.297522	0.0	0.0	0.0	773.565303

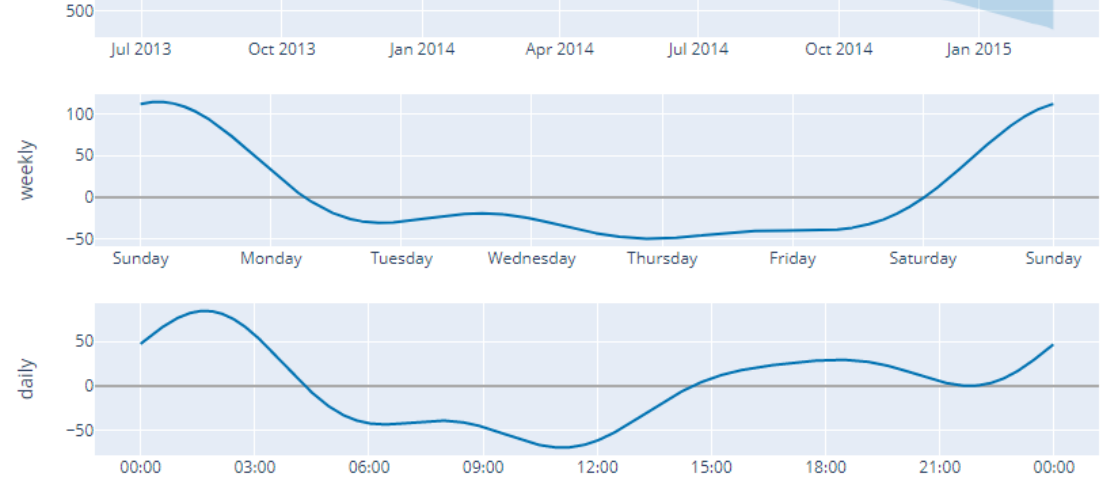
Out[14]:

	ds	yhat	yhat_lower	yhat_upper
39588	2015-02-14 13:10:00	886.791103	-877.723277	2561.838850
39589	2015-02-15 13:10:00	907.572781	-759.310288	2643.808433
39590	2015-02-16 13:10:00	794.209831	-867.498784	2609.175552
39591	2015-02-17 13:10:00	798.377558	-1032.077698	2586.746672
39592	2015-02-18 13:10:00	773.565303	-817.890385	2599.409994






1w 1m 6m 1y all
y
ds


trend
1000
500
Jul 2013
Oct 2013
Jan 2014
Apr 2014
Jul 2014
Oct 2014
Jan 2015
weekly
100
50
0
−50
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
daily
50
0
−50
00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00

REFERENCES

[1] Md Doulotuzzaman Xames, Jannatul Shefa, Ferdous Sarwar, "Bicycle industry as a post-pandemic green recovery driver in an emerging economy: a SWOT analysis", Springer-Verlag GmbH Germany, part of Springer Nature 2022 (Environmental Science and Pollution Research), July 2022.

[2] Weiwei Jiang, "Bike sharing usage prediction with deep learning: a survey", Springer-Verlag London Ltd., part of Springer Nature 2022, Neural Computing and Applications (2022) 34:15369–15385, June 2022.

[3] Suzana Regina Moro, Paulo Augusto Cauchick-Miguel, "An Analysis of a Bike-Sharing System from a Business Model Perspective", Brazilian Journal of Operations & Production Management, Vol. 19, No. 2, e20221400, 2022, ISSN 2237-8960 (Online), June 2022.

[4] Yuanyuan Guo, Linchuan Yang, Yang Chen, "Bike Share Usage and the Built Environment: A Review", Frontiers in Public Health (www.frontiersin.org), Volume 10, Article 848169, February 2022.

[5] Songhua Hu, Mingyang Chen, Yuan Jiang, Wei Sun, Chenfeng Xiong, "Examining factors associated with bike-and-ride (BnR) activities around metro stations in large-scale dockless bikesharing systems", Journal of Transport Geography 98 (2022) 103271, Elsevier Ltd., December 2021.

[6] Hanning Song, Gaofeng Yin, Xihong Wan, Min Guo, Zhancai Xie, Jiafeng Gu, "Increasing Bike-Sharing Users' Willingness to Pay — A Study of China Based on Perceived Value Theory and Structural Equation Model", Frontiers in Psychology (www.frontiersin.org), Volume 12 | Article 747462, January 2022.

[7] Xiaonan Zhang, Jianjun Wang, Xueqin Long, Weijia Li, "Understanding the intention to use bike-sharing system: A case study in Xi'an, China", PLoS ONE 16(12): e0258790, December 2021.

[8] Puneeth B. R., Nethravathi P. S., "Bicycle Industry in India and its Challenges – A Case Study", International Journal of Case Studies in Business, IT, and Education (IJCSBE), 5(2), 62-74, ISSN: 2581-6942, Vol. 5, No. 2, August 2021.

[9] Vitória Albuquerque, Miguel Sales Dias, Fernando Bacao, "Machine Learning Approaches to Bike-Sharing Systems: A Systematic Literature Review", International Journal of Geo-Information, ISPRS Int. J. Geo-Inf. 2021, 10, 62, February 2021.

[10] Anil Jain, Nirmala Joshi, Anand J Mayee, "Factors motivating buying behavior of female two wheeler users in the district of Palghar", Journal of Management Research and Analysis, October-December, 2020;7(4):154-158, December 2020.

[11] S. Diwakar Raj, Dr. N. Kannan, "Factors Influencing Purchase of Two Wheeler - A Study with Reference to Chennai City", International Journal of Management, 11(12), 2020, pp 2977-2982, ISSN Print: 0976-6502 and ISSN Online: 0976-6510, December 2020.

[12] Gyugeun Yoon, Joseph Y.J. Chow, "Unlimited-ride bike-share pass pricing revenue management for casual riders using only public data", International Journal of Transportation Science and Technology 9 (2020) 159–169, January 2020.

[13] Leonardo Caggiani, Rosalia Camporeale, Branka Dimitrijević, Milorad Vidović, "An approach to modeling bike-sharing systems based on spatial equity concept", AIIT 2nd International Congress on Transport Infrastructure and Systems in a changing world (TIS ROMA 2019), 23rd-24th September 2019, Rome, Italy, Elsevier B.V., 2020.

[14] Mohammed Hamad Almannaa, "Optimizing Bike Sharing Systems: Dynamic Prediction Using Machine Learning and Statistical Techniques and Rebalancing", DOI: 10.13140/RG.2.2.26034.43202, Thesis for: PhD, Advisor: Hesham Rakha, Project: Bike Research, April 2019.

[15] Elisabete Arsenio, Elisabete Arsenio, Sofia Azeredo Lopes, Helena Iglésias Pereira, "Assessing the market potential of electric bicycles and ICT for low carbon school travel: a case study in the Smart City of ÁGUEDA", European Transport Research Review (2018) 10: 13, Springer, January 2018.

[16] Miriam Ricci, "Bike sharing: A review of evidence on impacts and processes of implementation and operation", Research in Transportation Business & Management 15 (2015) 28–38, Elsevier Ltd., April 2015.

[17] Angela Au, "Social Media Strategies Used in Marketing Custom Bicycle Framebuilding Companies", Doctoral Study - Walden University ScholarWorks (Walden Dissertations and Doctoral Studies Collection), November 2015.

[18] Inês Frade, Anabela Ribeiro, "Bicycle sharing systems demand", EWGT2013 – 16th Meeting of the EURO Working Group on Transportation, Procedia - Social and Behavioral Sciences 111 (2014) 518 – 527, Elsevier Ltd., February 2014.

[19] Darren Buck, Ralph Buehler, Patricia Happ, Bradley Rawls, Payton Chung, Natalie Borecki, "Are Bikeshare Users Different from Regular Cyclists? A First Look at Short-Term Users, Annual Members, and Area Cyclists in the Washington, DC Region", Transportation Research Record Journal of the Transportation Research Board 2387(-1):112-119, DOI: 10.3141/2387-13, December 2013.

[20] Carlos M. Vallez, Mario Castro, David Contreras, "Challenges and Opportunities in Dock-Based Bike-Sharing Rebalancing: A Systematic Review", Sustainability 2021, 13, 1829. https://doi.org/10.3390/su13041829, MDPI, February 2021.