CAB FARE PREDICTION

A Project Report submitted in the partial fulfillment of the

Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

S.Sushma Priyanka (19471A05O9)

G.Sai Roshini (19471A05L8)

SD.Sabeera Begum (20475A0517)

Under the esteemed guidance of

Dr.S.N.Tirumala Rao, M,Tech, Ph.D

H.O.D,CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS**)**

Accredited by NAAC with A+ Grade and NBA under Cycle-
1Approvedby AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDAROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "**CAB FARE PREDICTION**" is a bonafide work done by the team S.Sushma Priyanka(19471A05O9), G.Sai Roshini(19471A05L8), SD.Sabeera Begum(20475A0517) In partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** In the Department of **COMPUTER SCIENCE AND ENGINEERING** during 2022-2023.

PROJECT GUIDE

Dr. S.N.Tirumala Rao M.Tech, Ph.D

Head of the Department

PROJECT CO-ORDINATOR

Dr. M.Sireesha MTech, Ph.D

Assoc.Professor

HEAD OF THE DEPARTMENT

Dr. S.N.Tirumala Rao M.Tech, Ph.D

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the complete on of the project.We are extremely thank ful to our beloved chairman sir **M.V.Koteswara Rao** **B.Sc**who took keen interest in us in every effort throughout this course. We oweout sincere gratitude to our beloved principal **Dr.M.Sreenivasa Kumar** **M.Tech., Ph.D., MISTE., FIE(I).,**for showing hisk in and attention and valuable guidance through out the course.

We express our deep felt gratitude towards **Dr.S.N.Tirumala Rao** **M.Tech.,Ph.D.,** HOD ofCSE department and also our guide of CSE department whose valuable guidance and unstinting encourage mentenable us to accomplish our projects uccessfully intime.

We extend our sincere thanks towards **Dr.M.Sireesha** **M.Tech,Ph.D,** Associate professor &Project coordinator of the project for extending her encouragement. Their profound knowledgeandwillingnesshavebeenaconstantsourceofinspirationforusthroughoutthisprojectwork.

We extend our sincere thanks to all other teaching and non-teaching staff to departmentfortheircooperationandencouragementduringourB.Techdegree.

We have now ordstoacknowledge the warm affection ,constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

S.SushmaPriyanka (19471A05O9)
G.Sai Roshini (19471A05L8)
SD.Sabeera Begum (20475A0517)

ABSTRACT

Cab rides paint a vibrant picture of life in the city. The millions of rides taken each month can provide insight into traffic patterns, road blockage, or large-scale events that attract many people. With ride sharing apps gaining popularity, it is increasingly important for cab companies to provide visibility to their estimated fare since the competing apps provide these metrics upfront. Predicting fare of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example. Furthermore, this visibility into fare will attract customers during times when ridesharing services are implementing surge pricing. In order to predict fare, only data which would be available at the beginning farid was used.

This includes pick up and drop off coordinates, trip distance, start time, number of passengers. Regression models were used to predict fare amount. Now it becomes really important to manage their data properly to come up with new business ideas to get best results. In this scenario, earn most revenues. So, it becomes really important estimate the fare prices accurately.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineeringsciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature

CO425.4: Design and Modularize the project

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		✓											✓		
C425.2	✓		✓		✓								✓		
C425.3				✓		✓	✓	✓					✓		
C425.4			✓			✓	✓	✓					✓	✓	
C425.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C425.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**
2. **Middle level**
3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a **smart bottle for health care using sensors.**	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into **five modules**	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our four members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the **hospital management and in future updates in our project can be done based on air bubbles occurring in liquid in saline.**	PO4, PO7
CC4.2.8 CC4.2.	**The physical design includes hardware components like sensors, gsm module, software and Arduino.**	PO5, PO6

INDEX

LIST OF FIGURES

1.INTRODUCTION

1.1 INTRODUCTION

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly. Now a day's cab services are expanding with the multiplier rate. Cab Fare Prediction had a great deal of attention in present research. The ease of using the services and flexibility gives their customer a great experience with competitive prices.

Attributes used in the predicting cab fare are below mentioned:

Trip distance	**:**	If the distance to be traveled is more, then fare should be higher.
Time of Travel	**:**	During peak traffic hours, the taxi fare may be higher.
Day of Travel	:	Fare amount may differ on weekday and weekends
Weather Conditions	:	If it is snowing, there may be lower availability of cabs and hence higher fares
Is it a trip to/from airport	:	Trips to/from airport generally have a fixed fare.
Pickup or Drop-off Neighborhood	**:**	Fare may be different based on the kind of neighborhood.
Availability of taxi	:	If a particular location has a lot of cabs available, the fares may be lower.

1.2 EXISTING SYSTEM

One of the major problems of travel, and transportation is solved by taxi ridesharing. According to a recent report, most passengers approved of taxi ridesharing. Particularly, the most common ways passengers use ridesharing are by booking a ride on an App, or by taking ridesharing services offline through a taxi. The proposed system looks for routes while satisfying certain travel conditions. Stability of route here refers to the frequency of rerouting or transferring needs come up for a taxi driver. It could also be viewed as the probability of an unexpected road condition coming up and affecting the travel and time requirements of the driver and the passenger.

Disadvantages:

1. Doesn’t generate accurate and efficient results.
2. Computation time is very high.
3. Lacking of accuracy may result in lack of efficient further prediction.

1.3 PROPOSED SYSTEM

We proposed that we will find the days on which each basement has more trips. Can tap growing markets in suburban areas where predicting cab fares are not available. In this we have used supervised learning approach, because it suits the best as per the requirement of predictive analysis. Prediction is performed as data is collected from past, the model is trained to handle new data and predict the desired output.

Advantages:

1. Generates accurate and efficient results.
2. Computation time is greatly reduced.
3. Reduces manual work.
4. Efficient further prediction.

1.4. SYSTEM REQUIREMENTS

1.4.1 Hardware Requirements:

- System Type : Intel(R) Core ^{TM} 2 i7-5500U CPU @ 2.40GHz
- Cache memory : 4MB(Mega Byte)
- RAM : 8 gigabyte (GB)

1.4.2 Software Requirements:

- Operating System : Windows 10 Home, 64 bit Operating System
- Coding Language : Python,Flask
- Python distribution : Google Colab, PyCharm

2. LITERATURE SURVEY

2.1 MACHINE LEARNING

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly.

2.2 SOME MACHINE LEARNING METHODS

Machine learning algorithms are often categorized as supervised and unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- **Unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabelled data for training
  Typically a small amount of labeled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn't require additional resources.
- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and Error Search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior

within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 APPLICATIONS OF MACHINE LEARNING

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection

2.4 PREVALENCE OF PRICE PREDICTION

A raw dataset has abundant features, which makes entire dataset processing chaotic. Hence, Feature Extraction is applied to reduce the initial set of raw data into a more manageable form. Feature extraction also reduces redundant data for a given analysis. The merging of features forms a dataset that is able to contain most of the original information in summarized version.

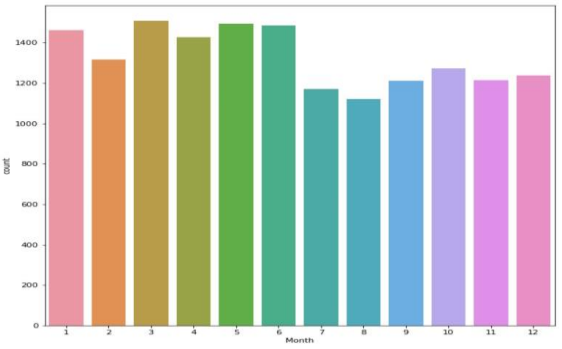


Fig: 2.4.1 Prevalence of Cab Fare Prediction

2.5 IMPORTANCE OF MACHINE LEARNING IN PRICE PREDICTION

Price forecasting is a useful feature for consumers as well as businesses. A price prediction tool motivates users to engage with a brand or evaluate offers in order to spend their money wisely. Price prediction enables businesses to set pricing in a manner that builds customer engagement and loyalty. With Machine Learning (ML) technology a price prediction problem is formulated as a regression analysis which is a statistical technique used to estimate the relationship between a dependent/target variable and single or multiple independent (interdependent) variables. In regression, the target variable is numeric.

2.6 DATA PRE-PROCESSING

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

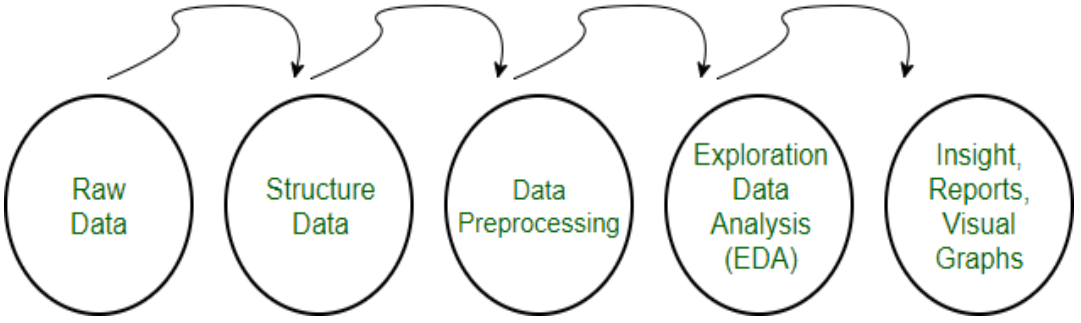


Fig: 2.6.1 Data Pre-processing

Need of Data Pre-processing

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format. For example, Random Forest algorithm does not support null

values; therefore to execute random forest algorithm null values have to be managed from the original raw data set. Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set and best out of them is chosen.

2.6.1 Principal Component Analysis:

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

2.6.2 Correlation:

This is part of an ongoing statistics-related blog post series in anticipation of cab fare prediction upcoming statistical tests resource

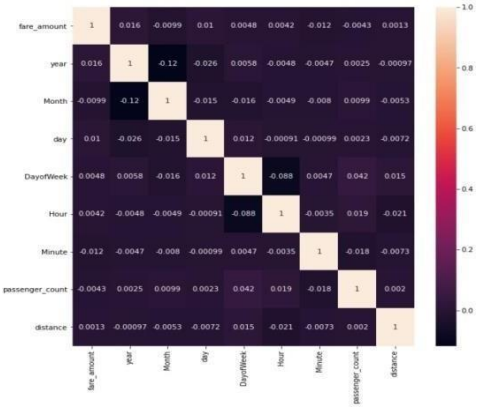


Fig: 2.6.2.1: Correlation for cab fare prediction

2.6.3 PCA with Correlation:

This PCA with correlation is using the correlation matrix which is equivalent to standardizing each of the variables (to mean 0 and standard deviation 1). In general, PCA with and without standardizing will give different results.

2.6.4 Missing values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value Nan. So that pandas can detect the missing values. The fig1 below is a heat map representing the missing values. In these graph missing values are present in features. We have filled that missing values using the median of the features. After filling the missing values our heat map looks like fig2.

```
fare_amount          24
pickup_datetime       1
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
passenger_count      55
dtype: int64
```

Fig: 2.6.4.1 Missing data visualization

2.6.5 Discretization

Conversion of a continuous-valued variable to a discrete value by creating a set of contiguous intervals that falls in the range of the variable's values is called discretization. We have categorized the using the following table. A classification algorithm such as Random Forests (RF) is chosen for its ability to handle high- dimensional data, benefits from discretization in the analysis of genetic data. Decision Tree is able to deal with both continuous and categorical attributes for classification. However, this technique handles the continuous attributes by discretization. We are using lasso since our data contains both continuous and categorical attributes we should use dummies.

2.6.6 Feature scaling

It is nothing but data normalization in data processing used to standardize the range of independent variables. This feature is very useful in data pre-processing step. The Standard Scalar will assume that the data is normally distributed within each attribute and will scale them in such a way that the distribution is now centered on 0, with a standard deviation is of 1. The mean and standard deviation are calculated for the feature and then the feature is scaled based on:

Yi–mean(y)/stdev(y)

2.6.7 Correlation coefficient method

We can find dependency between two attributes p and q using Correlation coefficient method using the formula.

$$r_{p,q} = \sum(p_i - \bar{p})(q_i - \bar{q})/n\sigma_p\sigma_q$$

$$= \sum(p_i q_i) - n\bar{p}\,\bar{q}/n\sigma_p\sigma_q$$

n is the total number of patterns, p_i and q_i are respective values of p and q attributes in patterns i, $\bar{p}$ and $\bar{q}$ are respective mean values of p and q attributes, σ_p , σ_q are respective standard deviations values of p and q attributes. Generally, $-1 \leq r_{p,q} \leq +1$. If $r_{p,q} < 0$, then p and q are negatively correlated. If $r_{p,q} = 0$, then p and q are independent attributes and there is no correlation between them. If $r_{p,q} > 0$, then p and q are positively correlated.

We can drop the attributes that are having correlation coefficient value as 0 as it indicates that the variables are independent with respect to the prediction attribute.

In-order to choose n value for decision tree algorithm we use the error rate. We choose the k value where the error rate is low. Figure 3 represents error rate with respect to k value before applying correlation coefficient. Figure 4 represents error rate with respect to k value after applying correlation coefficient.

2.7 REGRESSION

It is a process of categorizing data into given classes. Its primary goal is to identify the class of our new data.

2.7.1 Machine learning algorithms for regression

Research on data mining has led to the formulation of several data mining algorithms. These algorithms can be directly used on a dataset for creating some models or to draw vital conclusions and inferences from that dataset. Some popular data mining algorithms are Decision tree Regression, Linear Regression, Random Forest Regression, Ridge Regression, Lasso Regression, Principal Component Analysis, correlation, PCA with Correlation etc.

1. Decision Tree Regression:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

1. Discrete output example: A weather prediction model that predicts whether or not there'll be rain in a particular day.
2. Continuous output example: A profit prediction model that states the probable profit that can be generated from the sale of a product

2. Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression:

$$y = \theta_1 + \theta_2.x$$

While training the model we are given:

x: input training data (univariate – one input variable(parameter))

y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best $\theta 1$ and $\theta 2$ values.

$\theta 1$: intercept

$\theta 2$: coefficient of x

Once we find the best $\theta 1$ and $\theta 2$ values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

How to update θ1 and θ2 values to get the best fit line Cost Function (J):

By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the $\theta 1$ and $\theta 2$ values, to reach the best value that minimize the error between predicted y value

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

(prediction) and true y value (y).

Cost function (J) of Linear Regression is the **Root Mean Squared Error (RMSE)** between predicted y value (prediction) and true y value (y).

3. Random Forest Regression:

Random Forests are an ensemble learning method (also thought of as a form of nearest neighbor predictor) for classification and regression techniques. It builds multiple decision trees and then merges them together in-order to get more accurate and stable predictions. It constructs a number of Decision trees at training time and outputs the class that is the mode of the classes output by individual trees. It also tries to minimize the problems of high variance and high bias

by averaging to find a natural balance between the two extremes. Both R and Python have robust packages to implement this algorithm.

4. Ridge Regression:

The Ridge regression is a technique which is specialized to analyze multiple regression data which is multicollinearity in nature. Though linear regression and logistic regression are the most beloved members of the regression family, according to a record-talk, you must be familiar with using regression without regularization. Ridge regression is one of the most fundamental regularization techniques which is not used by many due to the complex science behind it. if you have an overall idea about the concept of multiple regressions, it's not so difficult to explore the science behind the Ridge regression .Regression is the same, what makes regularization different is that the way how the model coefficients are determined.

5. Lasso Regression:

Lasso regression which is one of the regression models that are available to analyze the data. Further, the regression model is explained with an example and the formula is also listed for reference. **LASSO** stands for **Least Absolute Shrinkage and Selection Operator.** Lasso regression is one of the regularization a method that creates parsimonious models in the presence of large number of features, where large means either of the below two things:

1. Large enough to enhance the tendency of the model to over-fit. Minimum ten variables can cause over fitting.
2. Large enough to cause computational challenges. This situation can arise in case of millions or billions of features.

2.8 IMPLEMENTATION OF MACHINE LEARNING USING PYTHON

Python is a popular programming language. It was created in 1991 by Guido van Possum.
It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.It is possible to write Python in an Integrated Development Environment, such as Thonny,Pycharm,Net beans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self-verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used by individuals for their own projects.

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best things about Keras is that it allows for easy and fast prototyping.

PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language

Processing(NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar charts, etc.

2.9 MACHINE LEARNING PRODUCTS

Before looking at the data it is important to understand how the company expects to use and benefit from this model. This first brainstorming helps to determine how to frame the problem, what algorithms to select and measure the performance of each one.

Machine learning often employ the same methods and overlap with data mining significantly, but while ML focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data. This is the analysis step of knowledge discovery in databases (KDD) [1].

Data visualization helps to visualize the data easily using different graphs, charts, plots etc. So here it is represented using scatter plot, bar graph and histogram, so that data can be analyzed properly. All these analysis is done in JupyterLab [2].

After identifying the approach, the next step is preprocessing the data. Looking at data refers to exploring the data, refining the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis (EDA) [3].

It is important to infer some knowledge from the existing data and come up with more valuable information. As the dataset already has date time variable, further the year, month, day, weekday and hours are calculated that might have an effect on the fare and to further perform some EDA on the data. Also as the longitude, latitude points are there, the distance traveled per ride is easily

calculated to derive a relationship between the fare amount and the distance. To calculate the distance Haversine Distance formula is used and the distance in kilometers is found.

The Haversine formula [4] calculates the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation. A decision tree is a tree-like graph with nodes representing the place where an attribute is picked and queried; edges represent the answers to the query, and the leaves represent the actual output or class label. Decision trees are nonlinear [5]. Decision Tree algorithms are referred to as Classification and Regression Trees (CART) [6].

The quality of a regression model is how well its predictions match up against actual values, and Error metrics are used to judge the quality of a model, which enables us to compare regressions against other regressions with varied parameters [7]. A. Root Mean Squared Error (RMSE) The Root Mean Squared Error (RMSE) and R-Squared are used for dealing with time series forecasting and continuous variables. The RMSE indicates the absolute fit of the model to the data, whereas R-Squared is a relative measure of fit [1, 8]. RMSE must be compared with the dependent variable as RMSE.RMSE must be compared with the dependent variable as RMSE is in the same units as the dependent variable.

Most of these models are linear and are not able to deal with the asymmetric behavior in most real-world sales data. Some of the challenging factors like lack of historical data, consumer-oriented markets face uncertain demands, and short life cycles of prediction methods results in inaccurate forecast.

3. SYSTEM ANALYSIS

3.1 SCOPE OF THE PROJECT

The scope of this system is to maintain cab ride details in datasets, train the model using model using the large quantity of data present in datasets and predict the Cab fare for the new test case by analyzing the given historical data.

3.2 ANALYSIS

The dataset that we use for the Cab fare prediction contain the details of following attributes to train and test the system for the prediction. . We have a total of 6 independent variables and 1 dependent variable. The dataset contains 16067 observations and 7 variables.

fare_amount – fare of the given cab ride

pickup_datetime - timestamp value indicating when the cab ride started.

pickup_longitude - float for longitude coordinate of where the cab ride

pickup_latitude - float for latitude coordinate of where the cab ride started.

dropoff_longitude - float for longitude coordinate of where the cab ride ended.

dropoff_latitude - float for latitude coordinate of where the cab ride ended.

passenger_count - an integer indicating the number of passengers in the cab ride.

Based on the above data, a regression model is required to guess how much should be the Fare amount for the given set of data.

	A	B	C	D	E	F	G
1	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
2	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.84161	40.712278	1
3	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1
4	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.76127	-73.991242	40.750562	2
5	7.7	2012-04-21 04:30:42 UTC	-73.98713	40.733143	-73.991567	40.758092	1
6	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1
7	12.1	2011-01-06 09:50:45 UTC	-74.000964	40.73163	-73.972892	40.758233	1
8	7.5	2012-11-20 20:35:00 UTC	-73.980002	40.751662	-73.973802	40.764842	1
9	16.5	2012-01-04 17:22:00 UTC	-73.9513	40.774138	-73.990095	40.751048	1
10		2012-12-03 13:10:00 UTC	-74.006462	40.726713	-73.993078	40.731628	1
11	8.9	2009-09-02 01:11:00 UTC	-73.980658	40.733873	-73.99154	40.758138	2
12	5.3	2012-04-08 07:30:50 UTC	-73.996335	40.737142	-73.980721	40.733559	1
13	5.5	2012-12-24 11:24:00 UTC	0	0	0	0	3
14	4.1	2009-11-06 01:04:03 UTC	-73.991601	40.744712	-73.983081	40.744682	2
15	7	2013-07-02 19:54:00 UTC	-74.00536	40.728867	-74.008913	40.710907	1
16	7.7	2011-04-05 17:11:05 UTC	-74.001821	40.737547	-73.99806	40.722788	2
17	5	2013-11-23 12:57:00 UTC	0	0	0	0	1
18	12.5	2014-02-19 07:22:00 UTC	-73.98643	40.760465	-73.98899	40.737075	1
19	5.3	2009-07-22 16:08:00 UTC	-73.98106	40.73769	-73.994177	40.728412	1
20	5.3	2010-07-07 14:52:00 UTC	-73.969505	40.784843	-73.958732	40.783357	1
21	4	2014-12-06 20:36:22 UTC	-73.979815	40.751902	-73.979446	40.755481	1

Fig: 3.2.1 Dataset of Cab fare prediction

3.3 DATA PREPROCESSING

Before feeding data to an algorithm we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values have to be managed using raw data.

Missing values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value Nan. So that pandas can detect the missing values. We treated the zero value as NaN and After the analysis of various imputation methods mean imputation was found to be accurate and so all the missing values were imputed with the mean value. After imputation it was found that there were no longer any missing values in our 'Data' dataset. After filling the missing values our heat map looks like fig

```
1 df.isnull().sum()
```

```
fare_amount          24
pickup_datetime       1
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
passenger_count      55
dtype: int64
```

Fig : 3.3.1 Missing Value Count

Outlier Analysis:

There are a lot of noisy data so it's important to clean the data for better model performance. All the outliers present in the dataset were assigned a value of NA and were later imputed with the mean.

Feature scaling:

It is nothing but data normalization in data processing used to standardize the range of independent variables. This feature is very useful in data pre-processing step. The Standard Scalar will assume that the data is normally distributed within each attribute and will scale them in such a way that the distribution is now centered on 0, with a standard deviation is of 1.

The mean and standard deviation are calculated for the feature and then the feature is scaled based on:

Yi–mean(y)/stdev(y)

Yi represents the values of attribute y

Correlation coefficient method:

We can find dependency between two attributes p and q using Correlation coefficient method using the formula.

$r_{p,q} = \sum(p_i - p)(q_i - q)/n\sigma p\sigma q$

$= \sum(p_i\, q_i) - npq/ n\sigma p\sigma q$

n is the total number of patterns, pi and qi are respective values of p and q attributes in patterns i, p and q are respective mean values of p and q attributes, $\sigma p, \sigma q$ are respective standard deviations values of p and q attributes. Generally, $-1 \leq r_{p,q} \leq +1$. If $r_{p,q} < 0$, then p and q are negatively correlated. If $r_{p,q}=0$, then p and q are independent attributes and there is no correlation between them. If $r_{p,q}>0$,then p and q are positively correlated.

We can drop the attributes that are having correlation coefficient value as 0 as it indicates that the variables are independent with respect to the prediction attribute. In-order to choose n value for KNN algorithm we use the error rate. We choose the k value where the error rate is low

3.4 CONFUSION MATRIX

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class.
A false positive (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the sales with outlets . TPR = tp / (tp + fn)

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.
TNR = tn / (tn + fp) =1-FPR

Precision or positive predictive value (PPV)

If the test result is positive what is the probability that the sales having its item id and outlet.
PPV = tp / (tp + fp)

Negative predictive value (NPV)

If the data doesn't having the outlet. NPV = tn / (tn + fn)

Miss rate or false negative rate (FNR)

It is the proportion of the individuals with a known positive condition for which the test result is negative.
FNR = fn / (fp + tn)

Fall-out or false positive rate (FPR)

It is the proportion of all the identified sales which doesn't have the outlets and contains outlets.
FPR = fp/ (fp + tn)

False discovery rate (FDR)

It is the proportion of all the identified sales which doesn't have the outlets and contains outlets.
FDR = fp / fp + tp

False omission rate (FOR)

It is the proportion of the individuals with a negative test result for which the true condition is positive.

FOR = fn / (fn + tn)

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified. ACC = (tp + tn) / (tp + tn + fp + fn)

F1 score

It is the harmonic mean of precision and sensitivity F1 = 2tp / (2tp+ fp + fn)

RMSE Score

Root mean square of the error that has occurred between the test values and the predicted values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|x_{predict} - x_{actual}|)^2}$$

4. DESIGN

The overall design model of our project looks like:

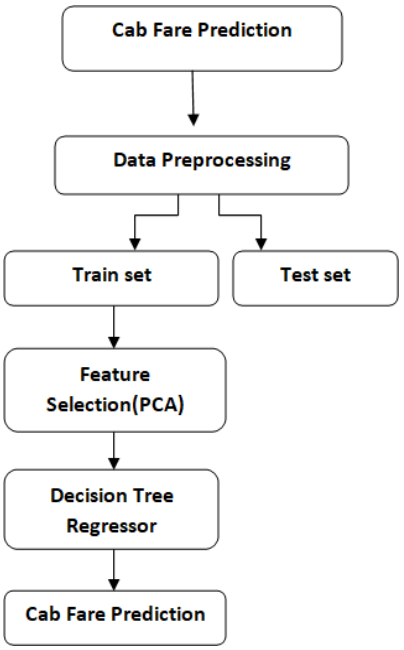


Fig 4.1: Design model of cab fare prediction

In our project, dataset is collected from Kaggle Repository and downloaded it into excel format. Later data pre-processing techniques like finding missing values and replacing them with mean etc. After that the dataset is divided into train set and test set and applied some feature selection techniques like Correlation, PCA and PCA with correlation. We identified PCA is the best solution to apply it on the train set. Finally, Decision Tree Regression is identified as best algorithms from all the algorithms we applied. Based on best algorithm Decision Tree Regression using PCA Technique, we can predict the fare of the cab ride.

5. IMPLEMENTATION

Step1 : Import libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import warnings
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.ensemble import GradientBoostingRegressor
```

Step2 : Load the data

```
df=pd.read_csv('cabdata.csv')
df.head()
df.shape
df.info()
```

Step3 : Data Preprocessing

```
cab['passenger_count'].value_counts()
cab['fare_amount'].sort_values(ascending=False)
cab['pickup_datetime']=pd.to_datetime(cab['pickup_datetime'],format='%Y-%m-
%d %H:%M:%S UTC',errors='coerce')
cab.head(10)
pip install missingno
import missingno as msno
(cab[['passenger_count','pickup_longitude','pickup_latitude','dropoff_longitude','dropoff_longitud
e','fare_amount']]==0).sum()
cab['fare_amount']=cab['fare_amount'].apply(pd.to_numeric,errors='coerce')
cab['fare_amount']=cab['fare_amount'].replace({0:np.nan})
cab['passenger_count']=cab['passenger_count'].fillna(0)
cab['passenger_count']=cab['passenger_count'].astype(int)
cab['passenger_count']=cab['passenger_count'].replace({0:np.nan})
cab['pickup_longitude']=cab['pickup_longitude'].replace({0:np.nan})
cab['pickup_latitude']=cab['pickup_latitude'].replace({0:np.nan})
cab['dropoff_longitude']=cab['dropoff_longitude'].replace({0:np.nan})
cab['dropoff_latitude']=cab['dropoff_latitude'].replace({0:np.nan})
cab.isnull()
cab.isnull().sum()
```

```
cab.describe()
cab['fare_amount']=cab['fare_amount'].fillna(cab['fare_amount'].mean())
cab['pickup_longitude']=cab['pickup_longitude'].fillna(cab['pickup_longitude'].mean())
cab['pickup_latitude']=cab['pickup_latitude'].fillna(cab['pickup_latitude'].mean())
cab['dropoff_longitude']=cab['dropoff_longitude'].fillna(cab['dropoff_longitude'].mean())
cab['dropoff_latitude']=cab['dropoff_latitude'].fillna(cab['dropoff_latitude'].mean())
cab=cab.dropna()
cab.isnull().sum()
msno.heatmap(cab)
cab.shape,cab.info()
cab.head()
convert_Data={'passenger_count':'int'}
cab=cab.astype(convert_Data)
cab.info()
```

Outliers

1.Fare Amount

```
cab.loc[cab['fare_amount']<0]
cab.loc[cab['fare_amount']<1,'fare_amount']=np.nan                # Fare Amount must not be nega
tive
cab=cab.dropna()
cab.shape
for i in range(1,11):
    print('passenger_count above'+str(i)+'={}'.format(sum(cab['passenger_count']>i)))
cab.loc[cab['passenger_count']>6]
cab[cab['passenger_count']<1]
cab=cab.drop(cab[cab['passenger_count']>6].index,axis=0)
sum(cab['passenger_count']>6),cab.shape
print('pickup_longitude above 180={}'.format(sum(cab['pickup_longitude']>180)))
print('pickup_longitude below -180={}'.format(sum(cab['pickup_longitude']<-180)))
print('pickup_latitude above 90={}'.format(sum(cab['pickup_latitude']>90)))
print('pickup_latitude below -90={}'.format(sum(cab['pickup_latitude']<-90)))
print('dropoff_longitude above 180={}'.format(sum(cab['dropoff_longitude']>180)))
print('dropoff_longitude below -180={}'.format(sum(cab['dropoff_longitude']<-180)))
print('dropoff_latitude below -90={}'.format(sum(cab['dropoff_latitude']<-90)))
print('dropoff_latitude above 90={}'.format(sum(cab['dropoff_latitude']>90)))
cab=cab.drop(cab[cab['pickup_latitude']>90].index,axis=0)
for i in ['pickup_longitude','pickup_latitude','dropoff_longitude','dropoff_latitude']:
    print(i,'equal to 0={}'.format(sum(cab[i]==0)))
cab.isnull().sum(),cab.shape,cab.dtypes
cab.describe()
```

Feature Selection

```
cab['year']=cab['pickup_datetime'].dt.year
cab['Month']=cab['pickup_datetime'].dt.month
cab['day']=cab['pickup_datetime'].dt.day
cab['DayofWeek']=cab['pickup_datetime'].dt.dayofweek
cab['Hour']=cab['pickup_datetime'].dt.hour
cab["Minute"]=cab['pickup_datetime'].dt.minute
plt.figure(figsize=(10,10))
```

```
sns.countplot(cab['year'])

plt.figure(figsize=(10,10))
sns.countplot(cab['Month'])

plt.figure(figsize=(10,10))
sns.countplot(cab['day'])

plt.figure(figsize=(10,10))
sns.countplot(cab['DayofWeek'])

plt.figure(figsize=(10,10))
sns.countplot(cab['Hour'])

plt.figure(figsize=(20,10))
sns.countplot(cab['Minute'])
cab.hist(bins=30, figsize=(15, 10))
cab.info(),cab.shape
```
Checking if number of passengers effecting fare:

```
plt.hist(cab['passenger_count'],color='green')
plt.figure(figsize=(10,5))
plt.scatter(x="passenger_count",y="fare_amount", data=cab,color='blue')
plt.xlabel('No. of passengers')
plt.ylabel('Fare_amount')
plt.show()
```

Checking if data effects the fare

```
plt.figure(figsize=(15,6))
plt.scatter(x="day",y="fare_amount",data=cab,color='blue')
plt.xlabel('Day')
plt.ylabel('Fare_amount')
plt.show()
plt.figure(figsize=(15,6))
plt.scatter(x="Month",y="fare_amount",data=cab,color='blue')
plt.xlabel('Month')
plt.ylabel('Fare_amount')
plt.show()
```

Checking if day of week effects the fare

```
plt.figure(figsize=(15,7))
plt.hist(cab['DayofWeek'], bins=100)
plt.xlabel('Day of Week')
plt.ylabel('Frequency')
plt.figure(figsize=(15,7))
plt.scatter(x=cab['DayofWeek'], y=cab['fare_amount'], s=50)
plt.xlabel('Day of Week')
plt.ylabel('Fare')
```

Checking if time effects the fare:

```
plt.hist(cab["Hour"])
plt.figure(figsize=(15,7))
cab.groupby(cab["Hour"])['Hour'].count().plot(kind="bar")
plt.show()
plt.figure(figsize=(10,5))
plt.scatter(x="Hour",y="fare_amount", data=cab,color='blue')
plt.xlabel('Hour')
plt.ylabel('Fare_amount')
plt.show()
plt.figure(figsize=(15,7))
cab.groupby(cab["Minute"])['Minute'].count().plot(kind="bar")
plt.show()
plt.figure(figsize=(10,5))
plt.scatter(x="Minute",y="fare_amount", data=cab,color='blue')
plt.xlabel('Minute')
plt.ylabel('Fare_amount')
plt.show()

from math import radians, cos, sin, asin, sqrt
def haversine(a):
    lon1=a[0]
    lat1=a[1]
    lon2=a[2]
    lat2=a[3]
    lon1,lat1,lon2,lat2=map(radians,[lon1, lat1, lon2, lat2])
                                                        #haversine formula
    dlon=lon2-lon1
    dlat=lat2-lat1
    a=sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c=2*asin(sqrt(a))
                                                    #Radius of earth in kilometers is 6371
    km=6371* c
    return km
cab['distance']=cab[['pickup_longitude','pickup_latitude','dropoff_longitude','dropoff_latitude']].a
pply(haversine,axis=1)
sum(cab['distance']==0)
cab=cab.drop(cab[cab['distance']==0].index,axis=0)
sum(cab['distance']==0)
```

Checking if distance effects the fare:

```
cab.sort_values(['distance','fare_amount'], ascending=False)
plt.figure(figsize=(10,5))
plt.scatter(x="distance",y="fare_amount", data=cab,color='blue')
plt.xlabel('distance')
plt.ylabel('Fare_Amount')
plt.show()
bins_0 = cab.loc[(cab['distance'] == 0), ['distance']]
bins_1 = cab.loc[(cab['distance'] > 0) & (cab['distance'] <= 10),['distance']]
```

```
bins_2 = cab.loc[(cab['distance'] > 10) & (cab['distance'] <= 50),['distance']]
bins_3 = cab.loc[(cab['distance'] > 50) & (cab['distance'] <= 100),['distance']]
bins_4 = cab.loc[(cab['distance'] > 100) & (cab['distance'] <= 200),['distance']]
bins_5 = cab.loc[(cab['distance'] > 200) & (cab['distance'] <= 300),['distance']]
bins_6 = cab.loc[(cab['distance'] > 300),['distance']]
bins_0['bins'] = '0'
bins_1['bins'] = '0-10'
bins_2['bins'] = '11-50'
bins_3['bins'] = '51-100'
bins_4['bins'] = '100-200'
bins_5['bins'] = '201-300'
bins_6['bins'] = '>300'
dist_bins =pd.concat([bins_0,bins_1,bins_2,bins_3,bins_4,bins_5,bins_6])
#len(dist_bins)
dist_bins.columns
plt.figure(figsize=(7,7))
plt.hist(dist_bins['bins'],bins=75)
plt.xlabel('Distance')
plt.ylabel('Frequency')
cab.shape
deletingthefeatures=['pickup_datetime','pickup_longitude','pickup_latitude','dropoff_longitude','dr
opoff_latitude']
cab=cab.drop(deletingthefeatures,axis=1)
cab['passenger_count']=cab['passenger_count'].astype('int64')

cab['year']=cab['year'].astype('int64')
cab['Month']=cab['Month'].astype('int64')
cab['day']=cab['day'].astype('int64')
cab['DayofWeek']=cab['DayofWeek'].astype('int64')
cab['Hour']=cab['Hour'].astype('int64')
cab['Minute']=cab['Minute'].astype('int64')


cab.dtypes
n=['fare_amount','year','Month','day','DayofWeek','Hour','Minute','passenger_count','distance']
cab_corr=cab.loc[:,n]
plt.figure(figsize=(10,10))
sns.heatmap(cab_corr.corr(),annot=True)
cab.head(20)
cab.shape,cab.info()
cab=cab[['fare_amount','passenger_count','year','Month','day','DayofWeek', 'Hour','Minute','distan
ce']]
cab.tail()
```

Step-4: Splitting as input and output

```
X=cab.drop('fare_amount',axis=1).values
Y=cab['fare_amount'].values
X.shape,Y.shape
X_train,X_test,y_train,y_test=train_test_split(cab.iloc[:,cab.columns != 'fare_amount'],cab.iloc[:,
0],test_size=0.30,random_state=1)
```

```
print(cab.shape,X_train.shape,y_train.shape,X_test.shape,y_test.shape)
from sklearn.feature_selection import SelectFromModel
sel_ = SelectFromModel(Lasso(alpha=0.005, random_state=0)) # remember to set the seed, the ra
ndom state in this function
sel_.fit(X_train,y_train)
sel_.get_support()
```

Step-5: Model development

1. Linear regression

```
model=LinearRegression()
model.fit(X_train,y_train)
pred_train=model.predict(X_train)
pred_test=model.predict(X_test)
RMSE_test=np.sqrt(mean_squared_error(y_test,pred_test))
RMSE_train= np.sqrt(mean_squared_error(y_train,pred_train))
print("Root Mean Squared Error For Train data = "+str(RMSE_train))
print("Root Mean Squared Error For Test data = "+str(RMSE_test))
#lrmodel.summary()
```

2. Random Forest Model

```
rf= RandomForestRegressor().fit(X_train,y_train)
pred_train_rf=rf.predict(X_train)
pred_test_rf=rf.predict(X_test)
RMSE_train_rf=np.sqrt(mean_squared_error(y_train, pred_train_rf))
RMSE_test_rf= np.sqrt(mean_squared_error(y_test, pred_test_rf))
print("Root Mean Squared Error For Training data = "+str(RMSE_train_rf))
print("Root Mean Squared Error For Test data = "+str(RMSE_test_rf))
print(r2_score(y_train, pred_train_rf))
print(r2_score(y_test, pred_test_rf))
rf.score(X_train,y_train)*100
```

3. Decision Tree Model

```
dt=DecisionTreeRegressor().fit(X_train,y_train)

pred_train_dt=dt.predict(X_train)
pred_test_dt=dt.predict(X_test)
RMSE_train_dt= np.sqrt(mean_squared_error(y_train, pred_train_dt))
RMSE_test_dt= np.sqrt(mean_squared_error(y_test, pred_test_dt))
print("Root Mean Squared Error For Training data = "+str(RMSE_train_dt))
print("Root Mean Squared Error For Test data = "+str(RMSE_test_dt))
r2_score(y_train, pred_train_dt),r2_score(y_test, pred_test_dt)
dt.score(X_train,y_train)*100
```

Pickle file

```
import pickle
with open("cabfare.pkl",'wb') as f:
pickle.dump(dt,f)
dt_model=pickle.load(open("cabfare.pkl","rb"))
```

Flask code to connect Frontend

```
from flask import Flask,render_template,request
import pickle
import numpy as np
import pandas as pd
import datetime
from sklearn.tree import DecisionTreeRegressor

dtr = pickle.load(open('cabfare.pkl','rb'))

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        data1 = request.form["day"]
        day = int(pd.to_datetime(data1, format="%Y-%m-%dT%H:%M").day)
        Month = int(pd.to_datetime(data1, format ="%Y-%m-%dT%H:%M").month)
        year= int(pd.to_datetime(data1, format ="%Y-%m-%dT%H:%M").year)
        Hour = int(pd.to_datetime(data1, format ="%Y-%m-%dT%H:%M").hour)
        Minute = int(pd.to_datetime(data1, format ="%Y-%m-%dT%H:%M").minute)
        data2=request.form["DayofWeek"]
        data3=request.form['passenger_count']
```

```
    data4=request.form['distance']
    data=[[day,Month,year,Hour,Minute,data2,data3,data4]]
    prediction=dtr.predict(data)[0]
  return render_template('index.html',prediction="Your cab price is {} ".format(prediction))

@app.route('/about.html')
def about():
  return render_template('about.html')

@app.route('/ContactUs.html')
def info():
  return render_template('ContactUs.html')

if__name__== "_main_":
  app.run(debug=True)
```

FrontEnd:

```
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cab Fare Prediction</title>
  <style>

    /* styling navlist */
    #navlist {

      position: absolute;
      width: 100%;
    }

    /* styling navlist anchor element */
   .ab1{
      float:left;
      display: block;
      color: #f2f2f2;
      text-align: center;
      padding: 25px;
      margin-left: 200px;
      text-decoration: none;
```

```
        font-size: 20px;
    }
    .ab2{

        display: block;
        color: #f2f2f2;
        text-align: center;
        padding: 25px;
        margin-left: 10px;
        text-decoration: none;
        font-size: 20px;

    }
    .navlist-right{
        float:right;
    }

    /* hover effect of navlist anchor element */
    .bar a:hover {
        background-color: #ddd;
        color: black;
    }
    .bg-container{
        background-image: url("https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQykND5hiJyWgMUR9SfTWwOPS1Sq3Z-
WUqADA&usqp=CAU");
        background-size:cover;
        background-repeat: no-repeat;
        height: 97vh;
        width: 98vw;

    }
    .btn1{
        margin-left:300px;
        display:inline-block;
    }
    .out{
        color:#ffffff;
    }
    .card{
        margin: 0px;
    }
    .heading{
        margin-left: 300px;
        margin-top: 10px;
        font-size:40px;
        color:black;
        font-family: sans-serif;
    }
    .heading a{
```

```
        color:white;
      }
      .res1{
        margin-left: 200px;
      }

  </style>


  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
      integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">



</head>
<script>
  function validateForm() {
    var x = document.forms["myForm"]["distance"].value;
    if (x == "" || x == null||x<0) {
      alert("Distance must be filled out and must be a positive value");
      return false;
    }
  }
  </script>

<body >

  <!-- <div id="navlist">
      <h1 class="heading"><a href="/">CAB FARE PREDICTION</a></h1>
      <div class="navlist-right">
      <a class="ab" href="/about.html">About</a>
      <a class="ab" href="ContactUs.html">Contact Us</a>
      </div>
  </div> -->
  <div class="bg-container">
      <div class="d-flex flex-row bar">
        <h1 class="heading"><a href="/">CAB FARE PREDICTION</a></h1>
        <a class="ab1" href="/about.html">About</a>
        <a class="ab2" href="ContactUs.html">Contact Us</a>
      </div>

      <br>


  <div class="ml-5">
      <form name="myForm" action="/predict" method="POST" onsubmit="return
validateForm()" required>
```

```
<div class="row">
  <div class="col-sm-3">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">DateTime</h5>
        <input type="datetime-local" name="day" id="day" required="required">
      </div>
    </div>
  </div>
  <br>
  <br>
  <br>
  <div class="col-sm-3">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">Day of the week</h5>
        <select name="DayofWeek" required="required">
          <option selected disabled value="">Enter Day of the week</option>
          <option value="0">0</option>
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
          <option value="4">4</option>
          <option value="5">5</option>
          <option value="6">6</option>
        </select>
      </div>
    </div>
  </div>
</div>
<br>
<br>
<div class="row">

  <div class="col-sm-3">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">Distance</h5>
        <input type="text" name="distance" placeholder="enter distance">
      </div>
    </div>
  </div>
<br>
<br>

  <div class="col-sm-3">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title">No of Passengers</h5>
          <select name="passenger_count" required="required" placeholder="enter
```

```
passenger_count">
                        <option selected disabled value="">No of Passengers</option>
                        <option value="1">1</option>
                        <option value="2">2</option>
                        <option value="3">3</option>
                        <option value="4">4</option>
                        <option value="5">5</option>
                        <option value="6">6</option>
                    </select>
                </div>
            </div>
        </div>
        </div>
            <br>
            <br>

            <div class="d-flex flex-row justify-content-start">
                <button class="btn btn-warning btn1">Get Fare</button>
            </div>
            <br>

    </form>

    <div class="res1">
        {% if prediction %}
        <h2 class="out">{{prediction}}</h2>
        {% endif %}
    </div>


    <br>
    <br>


</div>

</div>


<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
    crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
    integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"

crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
    integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
```

```
      crossorigin="anonymous"></script>


</body>
</html>

   html {
     box-sizing: border-box;
   }

   *, *:before, *:after {
     box-sizing: inherit;
   }

   .column {
     float: left;
     width: 25%;
     margin-bottom: 16px;
     padding: 0 18px;
   }

   .card {
     box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
     margin: 8px;
     padding-left: 10px;
   }

   .about-section {
     padding: 50px;
     text-align: center;
     background-color: #2b4c50;
     color: white;
   }

   .container {
     padding: 0 16px;
   }

   .container::after, .row::after {
     content: "";
     clear: both;
     display: table;
   }

   .title {
     color: rgb(150, 33, 33);
   }

   .button {
     border: none;
     outline: 0;
```

```
    display: inline-block;
    padding: 8px;
    color: white;
    background-color: #000;
    text-align:  center;
    cursor: pointer;
    width: 100%;
   }

   .button:hover {
    background-color: #555;
   }

   @media screen and (max-width: 650px) {
    .column {
     width: 100%;
     display: block;
    }
   }
</style>
</head>
<body>
 <div id="navlist">
  <a href="/">CAB FARE PREDICTION</a>
  <div class="navlist-right">
  <a href="/about.html">About</a>
  <a href="/ContactUs.html">Contact Us</a>
  </div>
</div>
<br>
<br>
<br>

<div class="about-section">
 <h1>Contact Us Page</h1>
</div>

<h2 style="text-align:center">Our Team</h2>
<div class="d-flex flex-row justify-content-center col-12">
 <div class="column">
  <div class="card">
   <div class="container">
    <br>
    <h2>Sushma Priyanka</h2>
    <p class="title">Project Leader</p>
    <p>Email id:priyankasushma13@gmail.com</p>
    <address>
    Contact<a href="mailto:priyankasushma13@gmail.com"> Sushma Priyanka</a>.<br>
     </address>
     <br>
     <br>
```

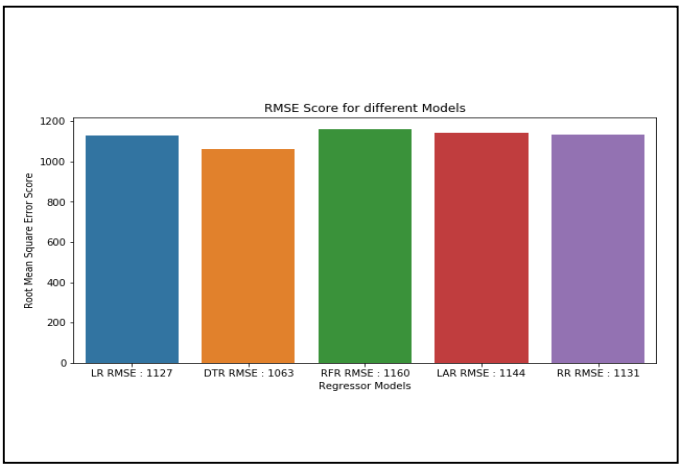
```
    </div>
  </div>
 </div>

 <div class="column">
  <div class="card">
   <div class="container">
    <br>
    <h2>Roshini</h2>
    <p class="title">Team Player</p>
    <p>Email id:roshinigrandhe@gmail.com</p>
    <address>
     Contact<a href="mailto:roshinigrandhe@gmail.com">Roshini</a>.<br>
       </address>
       <br>
       <br>
   </div>
  </div>
 </div>



 <div class="column">
  <div class="card">
   <div class="container">
    <br>
    <h2>Sabeera</h2>
    <p class="title">Team Player</p>
    <p>Email id:Sirishad@gmail.com</p>
    <address>
     Contact<a href="mailto:sirishad@gmail.com">Sabeera</a>.<br>
       </address>
       <br>
       <br>
   </div>
  </div>
 </div>
</div>
</body>
</html>

<!DOCTYPE html>
  <html>
  <head>
  <style>
    /* styling navlist */
    #navlist {
       background-color: #0f161d;
       position: absolute;
       width: 100%;
    }
```

```
    /* styling navlist anchor element */
    #navlist a {
       float:left;
       display: block;
       color: #f2f2f2;
       text-align: center;
       padding: 25px;
       text-decoration: none;
       font-size: 20px;
    }
    .navlist-right{
       float:right;
    }

    /* hover effect of navlist anchor element */
    #navlist a:hover {
       background-color: #ddd;
       color: black;
    }
  p,h1 {
    color:white;
    text-indent: 30px;
  }
  </style>
  </head>
  <body bgcolor=black>
    <div id="navlist">
       <a href="/">CAB FARE PREDICTION</a>
       <div class="navlist-right">
       <a href="/about.html">About</a>
       <a href="/ContactUs.html">Contact Us</a>
       </div>
    </div>

    <br>
    <br>
    <br>

   <br>

  <h1>About</h1><br>

  <p>      Now a day’s cab services are expanding with the multiplier rate. Cab Fare Prediction had a great deal of attention in present research. The ease of using the services and flexibility gives their customer a great experience with competitive prices.The millions of rides taken each month can provide insight into traffic patterns, road blockage, or large-scale events that attract many people. With ridesharing apps gaining popularity.Furthermore, the visibility into fare will attract customers during times when ridesharing services are implementing surge pricing.This includes trip distance, start time, number of passengers.So,it becomes really important estimate the fare prices accurately.
```

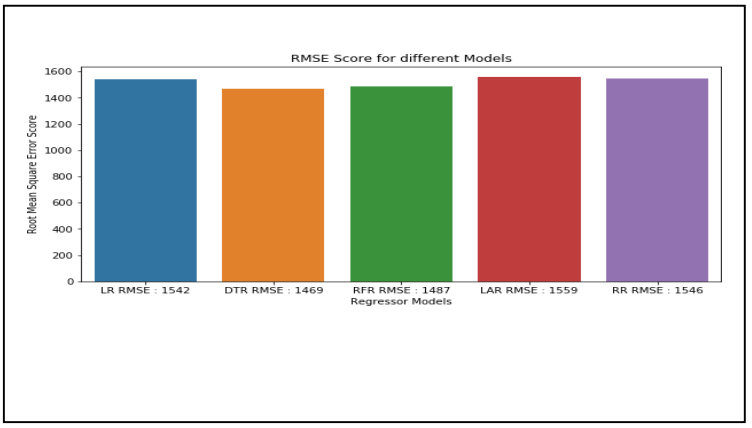
```
</p>

    </body>
    </html>
```

6. RESULT ANALYSIS

By considering the above Fig. 6.1 it is observed clearly that the RMSE of the Decision Tree Algorithm is low when compared to all the algorithms.



By considering the above Fig. 6.2 it is watched unmistakably that the RMSE of the Decision Tree Algorithm is low when contrasted with all calculations.



7. OUTPUT SCREENS

Fig.7.1 Output screen for Prediction page

Fig.7.2 Output screen for output of the cab fare prediction

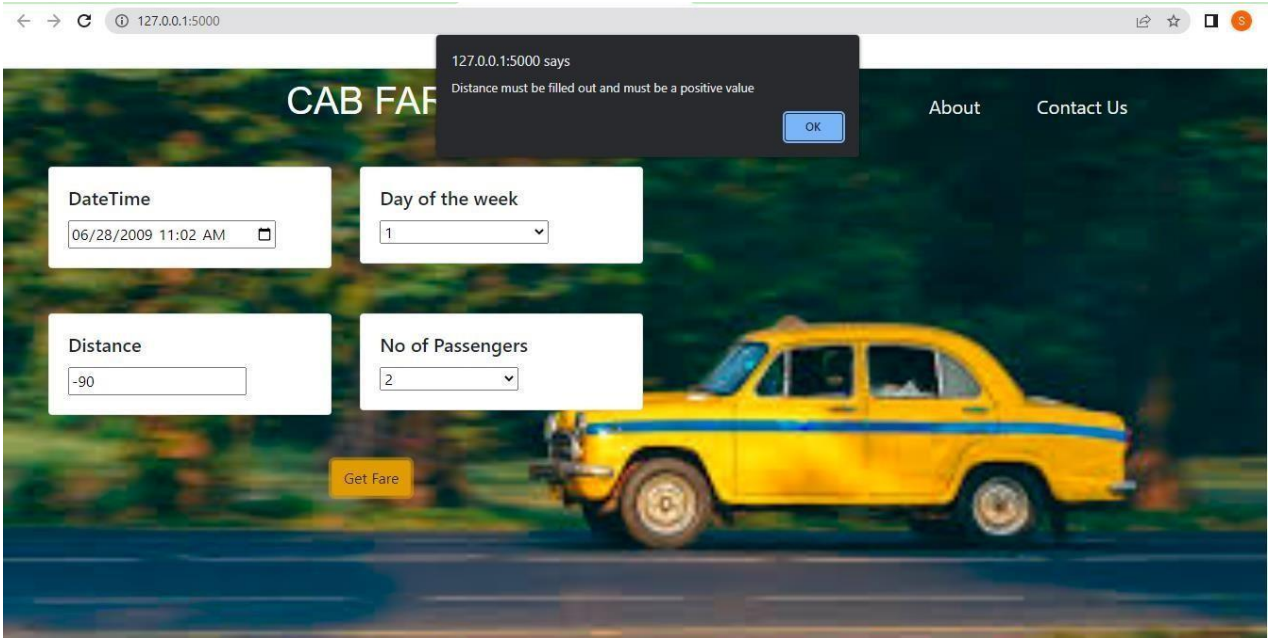


Fig.7.3 Output screen when distance is negative

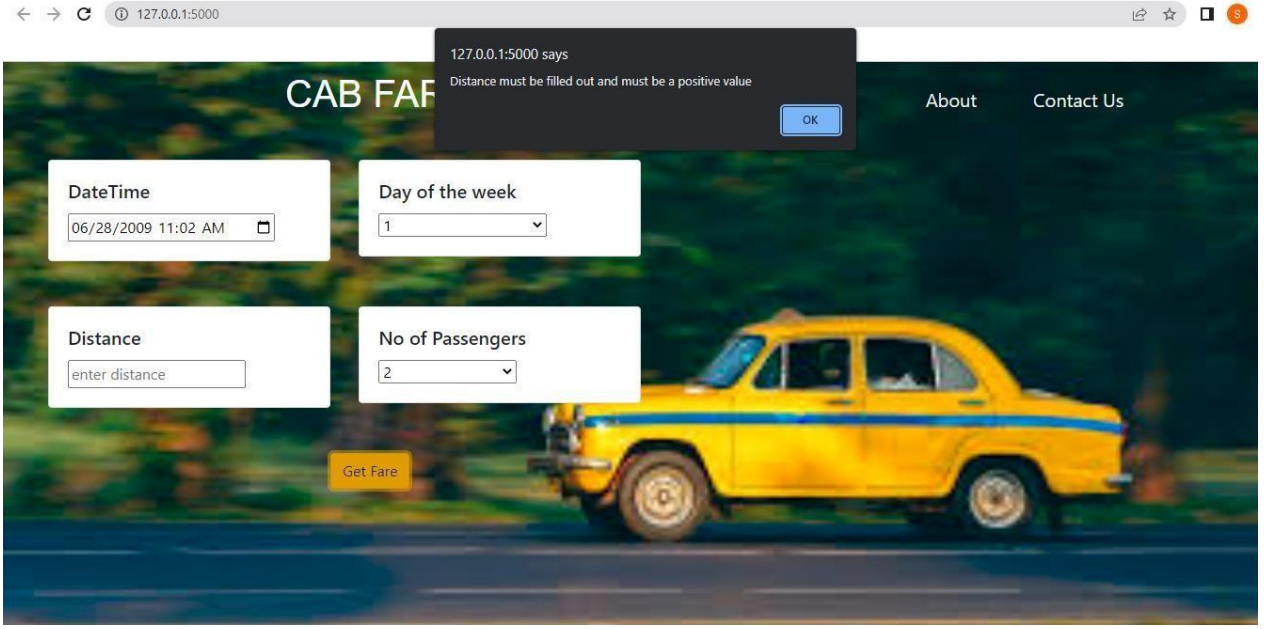


Fig.7.4 Output screen for distance is empty

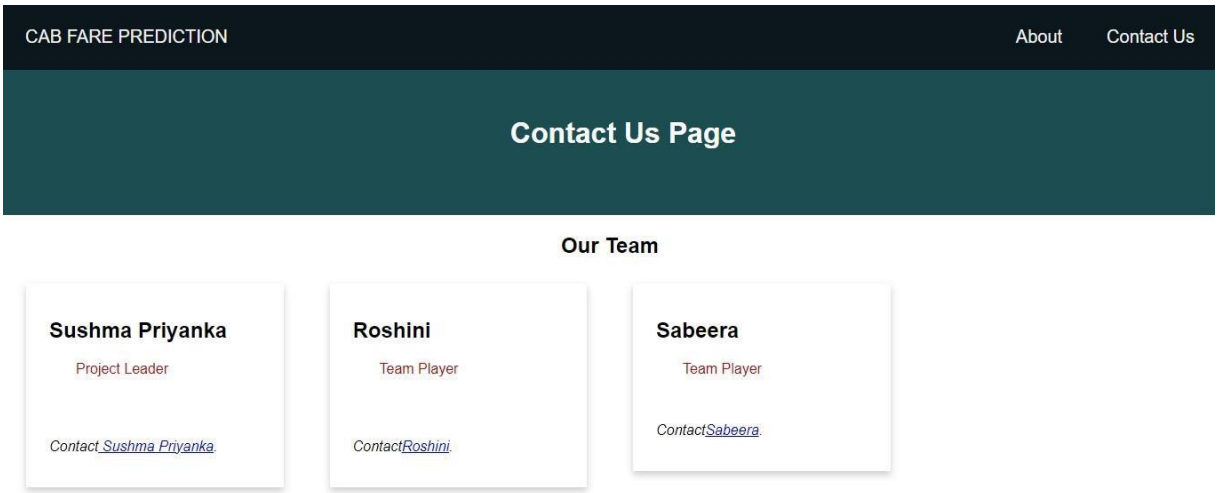


Fig.7.5 Output screen for Contact Us page

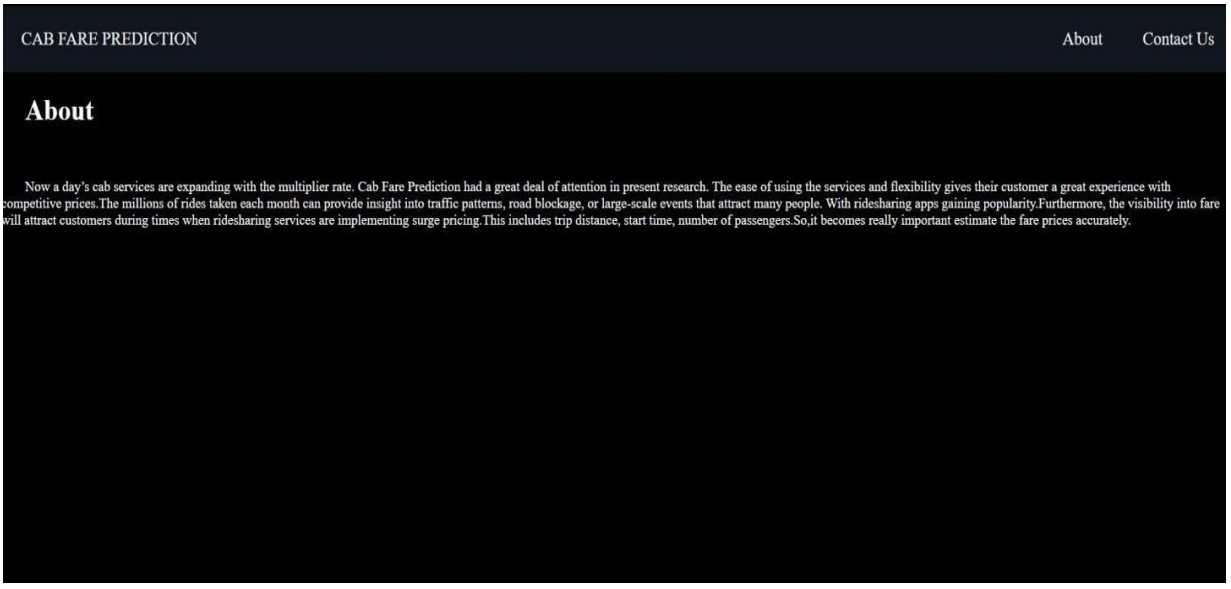


Fig 7.6: Output screen for about page

8. CONCLUSION & FUTURE SCOPE

After training and testing the results shown are fairly accurate. Decision tree is useful in regression as well as classification whereas linear regression helps to find the linear relation among the variables. Hence we reached to the conclusion that Decision tree is the best because it gives more accurate value as compared to other regression models. That is why Decision tree regressor algorithm is the best fit for the model selection as it has the lowest RMSE value and the highest R square value.

This project further can be developed as Android app. In future, the application can be enhanced by implementing decision tree forest algorithm in which we can find the correlation between every independent variable.

9.REFERENCES

[1] John D. Kelleher, Fundamentals Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies (The MIT Press), 1 st Edn.

[2] Data science work with public datasets, [Online] Available: **https://www.kaggle.com**,

[3] Chong Ho Yu, Exploratory data analysis in the context of data mining and re-sampling, International Journal of psychological research, 2010, vol.3, No.1

[4] Ruben Oliva Ramos, Jen Stirrup, Advanced Analytics with R and Tableau, Packt Publishing, 2017, ISBN: 9781786460110

[5] Machine Learning in Python,[Online] Available **https://scikit-learn.org/stable/**

[6] T. Divya and A. Sonali, "A survey on Data Mining approaches for Healthcare", International Journal of Biosciences and Bio-Technology, vol. 5, no. 5, (2013), pp. 241-266.

[7] Sebastian Raschka, Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, [Online] Available https://arxiv.org/abs/1811.12808,2016 [8] Weijie Wang and Yanmin Lu, Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model, ICMEMSCE, IOP Publishing, 324 (2018), doi:10.1088/1757-899X/324/1/0120

A Supervised Learning Algorithm for Cab Fare Prediction

S. Sushma Priyanka[1], G.V.L. Sai Roshini[2], SD.Sabeera Begum[3], S.N. Tirumala Rao[4]

[1,2,3]Student, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur (D.T) A.P, India

[4]Head of the Department, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur (D.T) A.P, India

priyankasushma13@gmail.com[1], roshinigrandhe@gmail.com[2], sabeerasayyad07@gmail.com[3], nagatirumalarao@gmail.com[4]

1. ABSTRACT- This is a predictive machine learning model. In this project, my objective was to predict the fare of a cab based on different factors mainly distance and time. I applied the process of data cleaning, data merging, and data visualization. Calculating the distance between the source and destination by using the haversine formula is the main key to predicting the cab fare. I implemented some ML regression algorithms and tested the score of the model, with that score we can finalize the machine learning model. Major apps like Ola, Uber, etc are introduced to cab rides in the major cities, but this research helps those who live in suburban areas. The study is based on supervised learning.

KEYWORDS: Supervised Learning, Outlier Analysis, Machine Learning, Price Prediction, Predictive Analysis.

2. INTRODUCTION

Machine learning is the process through which a computer may automatically learn from data, improve performance based on previous experiences, and generate predictions. Machine learning uses a variety of algorithms that work with enormous volumes of data. Data is used to training these algorithms, and after training, they create a model and perform a certain task. There are 3 different categories:

- Supervised learning: Throughout the machine learning phase, supervision is necessary. It contains both the input and the desired output, and the model is set up to forecast the desired outcome.
- Unsupervised learning: The model learns on its own by seeing patterns in the dataset without the need for supervision. Just input is provided, the model self-trains, and output results. Clustering and association are two examples.
- Reinforcement learning: The model is created utilizing the hit-and-miss approach in this type of learning. Its nature is reliant. Its input is the result of the previous operation. For instance, puzzle chess.

We used the supervised learning approach in this work because it best meets the requirements of predictive analysis.

Predictions are produced based on past data acquired, and when the model is taught to deal with novel input and anticipate the predicted result, predictions are generated based on fresh data.

Today's taxi services are growing at a multiplication pace. In the current study, there was a lot of focus on cab fare prediction. Customers get a terrific experience with affordable costs thanks to how simple and flexible the services are to use. the steps this strategy involves.

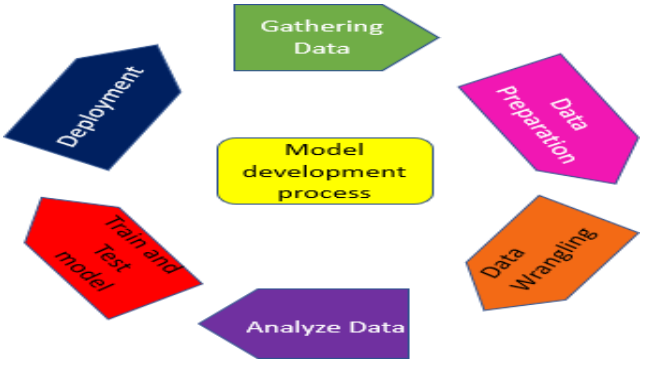


Fig.1 6 steps in model building

Many data mining algorithms have been developed as a result of data mining research. These algorithms may be applied directly to a dataset to build models or to derive important conclusions and inferences from it. Decision tree regression, linear regression, and random forest regression are some common data mining methods.

3. LITERATURE SURVEY

Understanding how the organization plans to use and benefit from this model is crucial before examining the data. This initial round of brainstorming aids in selecting the best algorithms, framing the challenge, and evaluating their effectiveness.

Although they frequently use the same techniques and have a lot in common, data mining and machine learning (ML) focus on different things. The goal of data mining is to discover previously undiscovered qualities in the data, whereas machine learning (ML) focuses on prediction utilizing prior attributes learned from training data. This is referred to as knowledge discovery (KDD) in databases [1].

Using various graphs, charts, plots, etc., data visualization makes it easier to understand the data. So that the data may be adequately examined, it is here depicted using a scatter plot, bar graph, and histogram. These analyses were all completed in JupyterLab [2].

The process of preparing the data comes after choosing a method. Data exploration, data rectification, and data visualization using graphs and charts are all terms used to describe the process of looking at data. A common name for this is Exploratory Data Analysis (EDA) [3].

4. PROPOSED SYSTEM

4.1 Data preprocessing

Typical noises, missing numbers, and sometimes an inappropriate arrangement make significant-world data unsuitable for machine-learning algorithms.

4.1.1 Dataset Description

The website of Kaggle, which has around 7 columns, was used for data gathering. These columns are the fee, the pickup and drop-off locations, the pickup and drop-off longitudes and latitudes, and the number of passengers. There were around 5 million rows of data in there as well. We used about 80,000 rows from that data, spanning the years 2009 to 2016. The following data view is provided:

	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	fare_amount
0	2015-01-27 13:08:24	-73.973320	40.763805	-73.981430	40.743835	1.0	15.01594
1	2015-01-27 13:08:24	-73.986862	40.719383	-73.998886	40.739201	1.0	15.01594
2	2011-10-08 11:53:44	-73.982524	40.751260	-73.979654	40.746139	1.0	15.01594
3	2012-12-01 21:12:12	-73.981160	40.767807	-73.990448	40.751635	1.0	15.01594
4	2012-12-01 21:12:12	-73.966046	40.789775	-73.988565	40.744427	1.0	15.01594

Fig 4.1.1. A sample of the dataset

The following list includes one target variable and six predictor variables:

Predictors:

1) Pickup Datetime: A timestamp value describing the commencement of the cab journey.
2) Pickup Longitude: A floating point number that represents the longitude coordinate at which the cab ride began.
3) Pickup Latitude: A floating point number that represents the latitude coordinate at which the cab ride began.
4) Dropoff Longitude: A floating point value indicating the longitude coordinate of the final location of the cab journey.
5) Dropoff Latitude: A floating point value indicating the latitude coordinate of the final location of the cab journey.
6) Number of Passengers: An integer value that represents the total number of passengers in the cab.

Goal: Price

4.1.2 Handling Missing Data

Missing Data can be handled in one of two ways after the dataset has been imported into the model:

A) By removing those specific rows

B) By substituting mean or median for those values

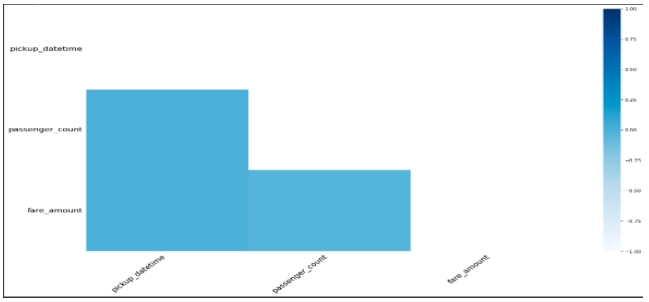


Fig 4.1.2(a): Visualization of missing values

We replaced missing values in our model with the mean of that particular column.

4.1.3 Outlier analysis

As there are many of noisy data, cleaning the data is crucial for improved model performance. In this instance, Turkey's technique is employed as a traditional method of eliminating outliers. We use box plots to show the outliers. Several insightful conclusions may be drawn from these plots, including the fact that each of the data sets has a significant number of outliers and extreme values.

	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	fare_amount
count	25821.000000	25821.000000	25821.000000	25821.000000	25821.000000	25821.000000
mean	-73.935899	40.713561	-73.932311	40.712394	1.657914	15.046575
std	2.071224	2.035526	2.111548	2.050426	1.271071	339.289571
min	-74.438233	-74.006893	-74.429332	-74.006377	1.000000	1.140000
25%	-73.992287	40.735397	-73.991210	40.734955	1.000000	7.300000
50%	-73.981977	40.752794	-73.980112	40.753738	1.000000	14.500000
75%	-73.967256	40.767287	-73.963780	40.768340	2.000000	15.015940
max	40.766125	41.709555	40.802437	41.696683	6.000000	54343.000000

Fig 4.1.3(a): refined data

The data is now refined as seen in Fig.4.1.3(a) after the outliers have been removed.

4.2 Correlation

The term "variable or attribute selection" is commonly used in machine learning. Choosing the traits that have the most influence on the aim variable is essentially what this procedure entails. The fare amount in this situation is the target or prediction variable, whilst the other factors are independent.

.

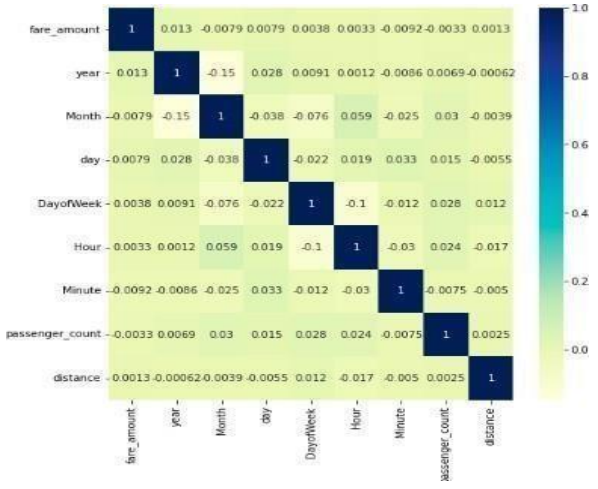


Fig 4.2(a): Correlation Heat Map

4.3 PCA

Principal component analysis (PCA) is a dimensionality reduction method that facilitates the identification of correlations and patterns in data collection.

4.4. Feature Engineering

To generate more useful information, it is crucial to draw certain conclusions from the available data. The distance covered in each ride may be simply estimated to establish a correlation between the fee amount and the distance since the longitude and latitude markers are present, determining the distance. It is crucial for navigational purposes.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25821 entries, 0 to 16065
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   pickup_datetime    25821 non-null  datetime64[ns]
 1   pickup_longitude   25821 non-null  float64
 2   pickup_latitude    25821 non-null  float64
 3   dropoff_longitude  25821 non-null  float64
 4   dropoff_latitude   25821 non-null  float64
 5   passenger_count    25821 non-null  int64
 6   fare_amount        25821 non-null  float64
 7   year               25821 non-null  int64
 8   Month              25821 non-null  int64
 9   day                25821 non-null  int64
 10  DayofWeek          25821 non-null  int64
 11  Hour               25821 non-null  int64
 12  Minute             25821 non-null  int64
dtypes: datetime64[ns](1), float64(5), int64(7)
memory usage: 3.8 MB
(None, (25821, 13))
```

Fig 4.4(a): Feature engineering

4.5 Model Selection

Modeling usually referred to as model selection is a crucial step that follows data pretreatment. The process of selecting a model from a wide pool of models to solve a prediction problem is known as model selection. Predicting the fare amount is our challenge. Regression is the issue here. The dependent variable in classification issues is categorical. Regression models will thus be used to analyze training data and make predictions about test data.

In this study, we employ a tree-based technique called Random Forest to address classification and regression issues. In regression models, the usage of a linear regression model is also common. The entire set of data was divided into two categories: train data (70%), and test data (30%).

4.5.1 Random Forest

Based on supervised learning, which supports both classification and regression, this model. Since it is composed of several decision trees that work together to forecast the goal value. A group of trees provides a value that is more accurate than one single tree.

4.5.2 Linear Regression

It is a supervised learning-based method. In light of independent factors, it provides the desired prediction value. It is mostly used to determine how dependent and independent variables are related.

Using the use of an accessible independent variable, To predict the value of a dependent variable, linear regression is performed. This strategy creates a linear relationship between the two groups of data a result. The following equation serves as the algorithm's foundation:

$$y=a+bx$$

Finding the appropriate values for a and b is the objective, with x acting as an independent variable and y acting as a dependent variable.

4.5.3 Decision tree

A decision tree is a graph that resembles a tree with nodes for the attributes that are selected and for which questions are asked, edges for the responses to those questions, and leaves for the actual output or class label. Nonlinearity describes decision trees. Classification and Regression Trees are two terms used to describe decision tree techniques.

4.6 Data Visualization

Using various graphs, charts, plots, etc., data visualization makes it easier to understand the data. So that the data may be adequately examined, it is here depicted using a scatter plot, bar graph, and histogram.

4.6.1 Histogram

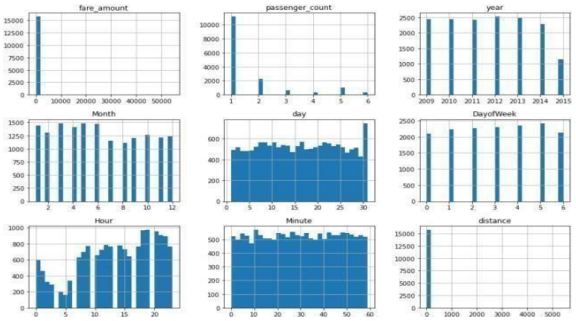


Fig 4.6.1 Histogram of all attributes of dataset

4.7 Proposed model

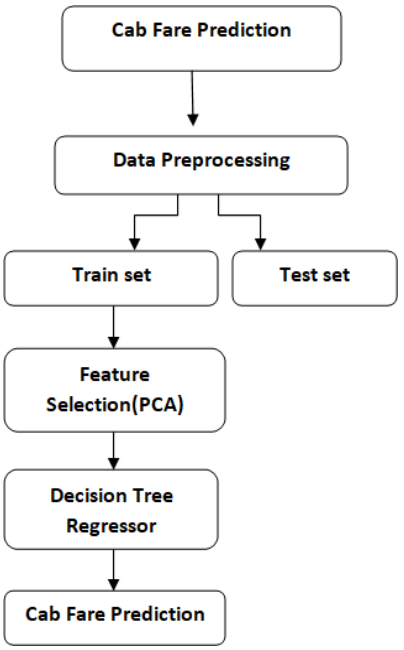


Fig 4.7(a): Design model of Cab fare prediction

5. RESULT ANALYSIS

The degree to which a regression model's predictions agree with observed values is a measure of its quality. Regressions may be compared to other regressions with various parameter values thanks to error metrics, which are used to evaluate a model's quality. We'll discuss particular regression error measures like -

- R square: The model is more accurate the higher the value. The values are converted to percentages ranging from 0 to 1.
- RMSE (Root Mean Square Error): The square root of the MSE error rate. Although the RMSE of 0.6 is modest, it is no longer that little. But the better, the lower the RMSE.

The model that is deemed to be the best and most accurate is also the one that has the highest R square and lowest RMSE values.

Algorithm	Accuracy
Linear regression	0%
Random forest	80%
Decision tree	100%

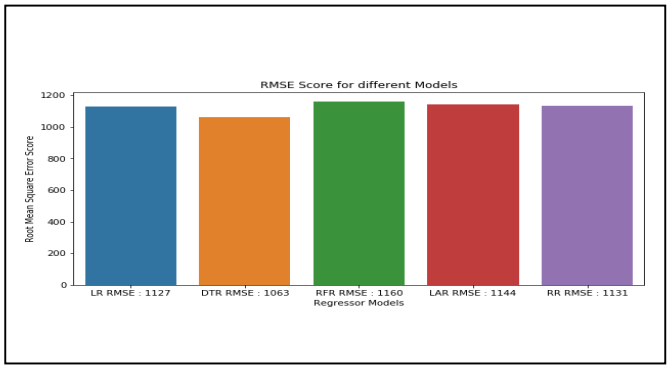


Fig 5(a): Comparison of algorithms with correlation

By considering the above Fig. 5(a) it is observed clearly that the RMSE of the Decision Tree Algorithm is low when compared to all the algorithms.

We are utilizing a decision tree method to forecast the cab fare using machine learning. This method can predict the result with high accuracy and less rmse value.

6. OUTPUT SCREENS

Fig 6(a): Cab fare prediction page

Using the above cab fare predictor, we predict the fareof the cab ride.

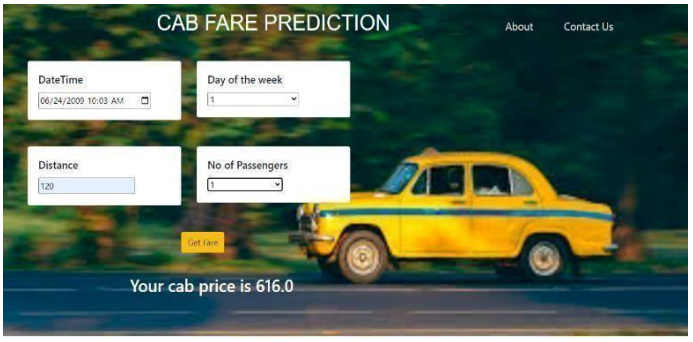


Fig 6(b): Output of cab fare prediction

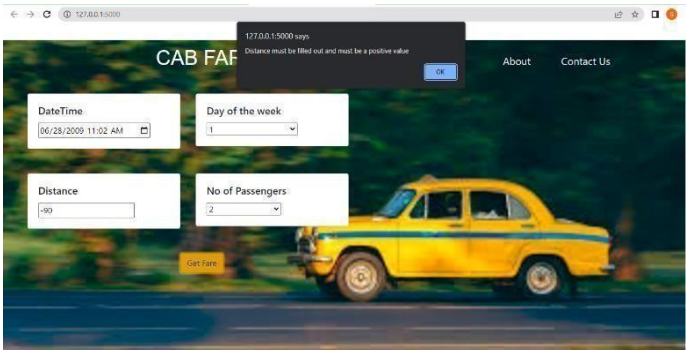


Fig 6(c): Output screen when distance is negative

7. CONCLUSION

The degree to which forecasts and observed values agree determines whether a regression model is successful. The dependent variable is continuous in issues involving regression. The dependent variable is categorical in classification issues. Problems with classification and regression may both be solved with Random Forest. In contrast to linear regression, decision trees do not use an equation to describe the connection between the independent and dependent variables. The decision tree is the most effective model out of the remaining three since it has the best R-Squared and RMSE scores, which measure how well the model fits the data and how much variability it explains.

8. REFERENCES

[1] John D. Kelleher, Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies (The MIT Press), 1st Edn.

[2] Data science work with public datasets, [Online] Available: https://www.kaggle.com,

[3] Chong Ho Yu, Exploratory data analysis in the context of data mining and re-sampling, International Journal of psychological research, 2010, vol.3, No.1

[4] Ruben Oliva Ramos, Jen Stirrup, Advanced Analytics with R and Tableau, Packt Publishing, 2017, ISBN: 9781786460110

[5] Machine Learning in Python, [Online] Available https://scikit-learn.org/stable/

[6] T. Divya and A. Sonali, "A survey on Data Mining approaches for Healthcare", International Journalof Biosciences and Bio-Technology, vol. 5, no. 5, (2013), pp. 241-266.

[7] Sebastian Raschka, Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, [Online] Available https://arxiv.org/abs/1811.12808,2016 .

DG4

ORIGINALITY REPORT

7%	4%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Purdue University Student Paper	1%
2	S. Siva Sunayna, S. N. Thirumala Rao, M. Sireesha. "Chapter 25 Performance Evaluation of Machine Learning Algorithms to Predict Breast Cancer", Springer Science and Business Media LLC, 2022 Publication	1%
3	Submitted to St. Xavier's College Student Paper	1%
4	www.ijraset.com Internet Source www.coursehero.com	1%
5	Internet Source mafiadoc.com	1%
6	Internet Source	1%
7	media.neliti.com Internet Source	<1%

8	Submitted to The Technical University of Kenya Student Paper	<1 %
9	www.mdpi.com Internet Source	<1 %
10	"Intelligent Information and Database Systems", Springer Science and Business Media LLC, 2017 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23054

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Somu Sushma Priyanka , Narasaraopeta Engineering College has presented the paper title A Supervised Learning Algorithm for Cab Fare Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23054

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Grandhe Venkata Lakshmi Sai Roshini , Narasaraopeta Engineering College has presented the paper title A Supervised Learning Algorithm for Cab Fare Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23054

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Sayyad Sabeera Begum , Narasaraopeta Engineering College has presented the paper title A Supervised Learning Algorithm for Cab Fare Prediction in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23015

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Dr.S.N.Tirumala Rao , Narasaraopeta Engineering College has presented the paper title Dog Breed Classification using Deep Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar