

AN EFFICIENT PARKING ALLOCATION SCHEME BASED ON K-MEANS CLUSTERING ALGORITHM

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Technology Degree in Information Technology

By

SMRITHI R - 38120079

NITHYA B - 38120054



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
JeppiaarNagar, RajivGandhiSalai, Chennai –
600119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **SMRITHI R(Reg.no.38120079)** and **NITHYA B(Reg.no.38120054)** who have done the Project work as a team who carried out the project entitled **"AN EFFICIENT PARKING ALLOCATION SCHEME BASED ON K-MEANS CLUSTERING ALGORITHM"** under my supervision from October 2021 to May 2022.

Internal Guide

(Name in Capital letters with signature)

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SMRITHI R, NITHYA B** hereby declare that the project report entitled **“AN EFFICIENT PARKING ALLOCATION SCHEME BASED ON K-MEANS CLUSTERING ALGORITHM”** done by us under the guidance of **Dr.BevisJinila.Y,M.E., Ph.D.**, at “Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119” is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering/Technology degree in Information Technology.

DATE: 07.04.2021

PLACE: Chennai

SIGNATUREOFTHECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing and Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. BEVISH JINILA.Y, M.E., Ph.D., for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all teaching and non-teaching staff members of the Department of INFORMATION TECHNOLOGY who was helpful in many ways for the completion of the project.

ABSTRACT

AN EFFICIENT PARKING ALLOCATION SCHEME BASED ON K-MEANS CLUSTERING ALGORITHM

With the development of the automobile sector in our country in recent years, the number of vehicles in use has constantly increased. As a result, the rate of expansion in the number of urban motor vehicles greatly outpaces the rate of parking facility building. The idea of ongoing improvement and development of parking distribution systems has always been important to the city's smooth operation. As a result, building an effective and dynamic parking algorithm will be a significant step forward in addressing the problem of urban parking shortages. As a result, utilising the K-Means Algorithm, this study developed a strategy for parking allocation. The findings of the experiments reveal that a dynamic parking distribution model based on K-Means Clustering can not only anticipate and allocate parking spaces in real time, but also enhance the utilisation rate of various types of parking spaces. As a result, it makes a substantial contribution to smarter and more sustainable urban parking management. In this paper an efficient parking allocation model is proposed and the research findings to predict the number of available parking spaces is discussed.

TABLE OF CONTENTS

CHAPTER NO	TITLE	NAME	NO
	PAGE		
	ABSTRACT		5
	LIST OF FIGURES		7
	LIST OF ABBREVIATION		8
1	INTRODUCTION9		
	1.1 GENERAL9		
	1.2 OUTLINE OF THE PROJECT11		
2	LITERATURE SURVEY12		
	2.1 LITERATURE SURVEY12		
3	METHODOLOGY14		
	3.1 PROGRAMMING LANGUAGES14		
	3.2 LIBRARIES AND FRAMEWORKS14		
	3.3 SYSTEM REQUIREMENTS15		
	3.4 IDE15		
	3.5 USER INTERFACE15		
	3.6 SOFTWARES USED16		
	3.7 ARCHITECTURE OF THE MODEL16		
	3.8 ALGORITHM USED18		
	3.9 WORKING19		
	3.10 PROJECT DESCRIPTION23		
4	RESULTANDDISCUSSION25		
	5.1 SYSTEM REQUIREMENT25		
	5.2 RESULT25		
	5.3 PERFORMANCE ANALYSIS26		
	CONCLUSION AND FUTURE WORK27		
5	5.1 CONCLUSION27		
	5.2 FUTURE WORK27		
	5.3 REFERENCES27		
	APPENDIX29		
	A.SCREENSHOTS29		

B.SOURCE CODE33
C.PUBLICATION WITH59
PLAGIARISM REPORT

LIST OF FIGURES

	FIG NO	FIG NAME	
	PAGENO		
3.1	ARCHITECTURE 15	OF THE MODEL	
3.2	REPRESENTATION 17	OF K-MEANS CLUSTERING	
		ALGORITHM	
3.3	USE CASE DIAGRAM 20	OF THE MODEL	
	3.4	WORKFLOW OF THE	21
		MODEL	
	4.1	RESULT	30
	4.2	PERFORMANCE	31
		ANALYSIS OF	
		VARIOUS ALGORITHMS	

LIST OF ABBREVIATIONS

ABBREVIATION

EXPANSION

IDE

Integrated Development

Environment

UI

User Interface

C#

C-Sharp

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Our Country with the rise in the development of economy, has seen an enormous increase in the automobiles. Because of this, the number of cars owned by the citizens of the country have been rising year by year. As a result of this, the issues related to parking have become persistent. With cities becoming busy everyday, and now that everyone has a vehicle parking related problems has become a part and parcel of life. Finding parking spaces has become challenging because of the lack of available parking spaces in the city, especially within the core areas of the city. As a result, Drivers spend more time in searching for available parking facilities, which causes traffic congestion, pollution and fuel consumption in the city. The existing system with respect to the parking is in no way helpful who are driving cars as it does not fulfill the need of finding an empty space and allocating the car. Hence as a relief to the people who are having a challenging time with finding a parking space, a dynamic machine learning model is introduced in this project, which is helpful in predicting the parking spots and plays a pivotal role in the allocation of cars in the given spots.

Today, machine learning algorithms play a huge role in automation. They are becoming developed every now and then in today's world. Gone were the days when it used to be manual method for finding parking slots. But now, based on the data given, the algorithm would be developed and the predictions would be made. This is the beauty of Machine Learning. Parking demand refers to the amount of parking that would be used at a certain time, place and duration. It is a function of vehicle ownership, trip rate, mode choice, duration, location and land-use, etc. Parking demand

has usually daily, weekly and even annual cycles. For example, parking demand peaks on weekdays at office-business areas while on weekends at market, restaurants, theatres, etc. Moreover, parking demand can change with transportation, land-use and demographic patterns. If a building change from residential to industrial or office-use, density and neighbourhood pattern may change, quality of transit service may change, all of them affect the parking demand and duration. Different types of trip purpose have different parking duration. Commuters need parking for larger duration and hence are relatively more price sensitive. An effective management of the parking demand especially in large central cities is an ongoing challenge as they face competing objectives and ever-increasing demands for space. As the main objective of the parking management is to balance the demand and supply, this study emphasized to model a parking duration in relation with the driver's socioeconomic and travel characteristics for two land-uses i.e., office-business (OB) and market/shopping (MS), to discover parameters which should be targeted at strategic level of planning.

In the modern era, population and economic growth in count with an increasing living standard of the people are to blame for rising number of automobiles. Ever-increasing number of private vehicles coupled with inefficient public transit system increasingly extracting the urban land for parking allocation which otherwise would overload the curb parking. Subsequently, it would reduce the traffic safety, air quality (reduced carriageway tends to increase congestion), etc. In critical phenomenon, when vehicle volume exceeds the network capacity, traffic congestion arises which has negative impacts on social as well as economic life of the society as a whole. The need for parking space is directly in nexus of demand associated with the upsurge in the vehicle plying on the roads. The challenges for parking are exacerbated as the parking supply in general is determined based on the parking necessity, lack in consideration of direct and indirect cost, now and in future as the parking spaces are built and convert the valuable lands to absorb the requirements. Various Algorithms are widely used for the statistical analysis requiring crisp input parameters.

Artificial Intelligence (AI) methods, especially those based on machine learning, are becoming essential for analysis where there is complexity in data and for decision support in various research areas including traffic and transportation.

Machine Learning Algorithms can be used for resembling the association of input and output signals of complex system and the synaptic weight matrix containing the knowledge of trained network cannot be easily decrypted. It easily describes the relationship it has with the data and does the predictions by the means of algorithm that is more accurate in predicting the data.

Thus, Parking is an inevitable issue in the fast-growing developing countries. Increasing number of vehicles require more and more urban land to be allocated for parking. However, a little attention has been conferred to the parking issues in developing countries like India. This study proposes a model for analysing the influence of car user's socioeconomic and travel characteristics on parking duration. Hence we are coming up with an algorithm that would best describe the predictions. Specifically, the K-Means Algorithm which we are dealing with, is deployed to capture the interrelationship between driver's characteristics and parking duration. They are highly efficient in learning and recognizing connections between parameters for best prediction of an outcome. Since, utility of K-Means has been used effectively, the study involves the best use of predictions and model interpretations. This Algorithm (K-Means Clustering Algorithm) shows the prediction for any car to be parked and also allocates the car based on the prediction being made. Therefore, the methodology can be adopted ubiquitously. Therefore This project could lead to enhanced parking policy and management to achieve the sustainability goals.

1.2 OUTLINE OF THE PROJECT

The project aims to make use of the automations taking place in the digital world right now, by building a machine learning model with the help of K-Means

Clustering Algorithm which helps in the prediction of the spaces which can get the car allocated and allocates the car by the means of the Algorithm implemented.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

This chapter discusses the merits and demerits of the works which are done already related to the proposed work. To Find a parking spot has always become an issue that should not be overlooked; it takes time and effort. Using data from the University of the Witwatersrand, we employed computer vision techniques to determine the state of the parking lot. Thus, it shows how to use Caffe and the Nvidia DiGITS framework to create a real-time parking space categorization system based on Convolutional Neural Networks (CNN) [1].

Ravi Kumar Gupta and Geetha Rani* et al.(2020) [2] titled 'Machine Learning and IoT based Real time parking system: Challenges and Implementation'. They have done a work on a car parking system using Machine Learning and IOT describing its Challenges and the way it is implemented. The methodology examined in this paper has accomplished the objective to develop a model which will predict the particular space for the car to be parked. The machine learning model used here helps us to easily understand the way it implements. This will help the Driver to park his/her car in an efficient manner. But considering the demerits into account, it is understood that the system does not help the Drivers to understand whether the given area is reserved or unreserved. Hence, there needs to be a way to locate the area as reserved or unreserved by any means of identification like giving some Colour which will indicate if the car parking space is available or not.

Predicting the location of parking is a long-standing issue that is crucial to our daily lives. Artificial neural networks are employed here to forecast the parking position,

which will assist drivers in deciding on a safe area to halt. As a result of this strategy, traffic on the highways becomes more familiar and comfortable, resulting in less turbulence. Reliability can be accomplished with a marginal error rate utilising the Deep Extreme Learning Machine (DELIM) technique [3], minimising the sceptical propensity. We used deep learning neural networks instead of feedforward and backward propagation in this study, the Proposed Car Parking Space Prediction (CPSP), to elaborate on the challenge of parking space for vehicles.

In this system there is a concept of Deep learning being used. Deep learning is an advanced type of Machine Learning and artificial Intelligence that imitates the way humans gain certain types of knowledge. In this paper, Convolutional Neural Network (CNN) [4] and various other machine learning algorithms are being used which yields good results. The idea behind the project is it helps to locate the free space by identifying its colour and with the help of convolutional layers being created. And this is considered to be the merit of this project. But when it comes to its demerits, it takes a long time in processing the input data given to it. That is because, deep learning is a study that takes input data for processing and would take huge amount of time in delivering the result.

Improved parking ability using various methodologies is discussed in [5-10]. The schemes on Intelligent Transportation Systems have inculcated various other sub domains like smart parking schemes to be employed to establish an efficient scheme. The schemes on smart cities and its deployment methods are discussed in [11-14].

CHAPTER 3

METHODOLOGY

3.1 PROGRAMMING LANGUAGES:

3.1.1 C#

C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework. C# can be used to create a number of different programs and applications like mobile apps, desktop apps, cloud based services, websites, enterprise software and games. It is a high level language.

3.2 LIBRARIES AND FRAMEWORKS

3.2.1. NET FRAMEWORK

It is a proprietary software framework developed by Microsoft that runs mainly on Microsoft Windows. It is used to develop software applications. It develops applications for web, windows and phone. This framework contains a large number of class libraries known as Framework Class Library (FCL). The software programs written in .NET are executed in the execution environment, which is called CLR (Common Language Runtime). These are the core and essential parts of the .NET framework.

Objectives of .NET Framework:

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.

2. To provide a code-execution environment to minimize software deployment and guarantee safe execution of code.

3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications. To make communication on distributed environment to ensure that code be accessed by the .NET Framework can integrate with any other code.

3.3 SYSTEM REQUIREMENTS:

3.3.1 HARDWARE REQUIREMENTS

- PROCESSOR: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
- RAM: 8GB
- SYSTEM TYPE: 64 bit Operating System
- STORAGE: 500 GB

3.3.2 SOFTWARE REQUIREMENTS:

- OPERATING SYSTEM: Windows 10
- FRONT END: VS(2017)
- BACK END: SQL SERVER(2017) R2

3.4 IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

IDE stands for Integrated Development Environment. It's a coding tool which allows you to write, test, and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools. And even though the IDE is a strictly defined concept, it's starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs. The IDE used in this project is Visual Studio.

3.5 USER INTERFACE

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

3.6 SOFTWARES USED

3.6.1 CODE EDITORS USED

VISUAL STUDIO

Microsoft Visual Studio is an integrated development Environment (IDE) from Microsoft. It is a creative launching pad that can be used to edit, debug and build code. It is even used for publishing an app. In our project's case it supports the language C#.

3.7 ARCHITECTURE OF THE MODEL

System architecture is a conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure and behaviour of the system.

Here, the system architecture shown in fig 3.1 represents the cars being parked on the given time slot. The first process is the machine learning algorithm called K-Means Algorithm is used for building the model. This is the algorithm which will take the input data and convert them into clusters. The next step is to build an efficient model which in order will predict the space according to the availability and helps the car to be allocated in a given space according to the clusters being obtained.

It uses the method of vector quantization which has originated from signal processing. K-means clustering focusses to split 'n' observations into k clusters in which every observation belongs to clusters having the mean which is nearest to it.



FIG 3.1 ARCHITECTURE OF THE MODEL

3.7.1 EXISTING SYSTEM:

- In this system, Allocation of cars has happened in the manual way.
- Human Errors are unavoidable and the cars will be allocated unevenly too.
- Integrated centralised management by humans in a manual way is possible.

PROPOSED SYSTEM:

- In the proposed system, an Automated machine learning model will allocate the cars.
- A degenerative technique of finding the free space and allocating based on the time slots and locations is done on this project.
- Here, Automation involves Accuracy, no manual intervention and preference and it is fast and efficient.

3.8ALGORITHM USED

K-MEANS CLUSTERING ALGORITHM

In machine learning, K-Means Clustering is understood to be an approach of unsupervised learning in the field of Machine Learning. Here it is used to address clustering problems. This algorithm does the job of separating 'n' observations into k clusters, by ensuring that each observation belonging to the cluster with the closest mean, which acts or serves as the leader of the cluster Here K describes the amount of pre-defined clusters. To go more in depth, it identifies the centroids present and does the allocation of all the data points present to the cluster near to it. Therefore, a wide range of cells emerge.

the unlabeled dataset into different clusters. K-means clustering aims to partition data into k clusters in a way that data points in the same cluster are similar and data points in the different clusters are farther apart. Similarity of two points is determined by the distance between them. It simply groups the data into different clusters which has similar data points.

The K-means clustering algorithm computes centroids and repeats until the optimal centroid is found. It is presumptively known how many clusters there are. It is also known as the flat clustering algorithm. The number of clusters found from data by the method is denoted by the letter 'K' in K-means.

In this method, data points are assigned to clusters in such a way that the sum of the squared distances between the data points and the centroid is as small as possible. It is essential to note that reduced diversity within clusters leads to more identical data points within the same cluster.

3.8.1 Working of K-Means Algorithm

The following stages will help us understand how the K-Means clustering technique works-

- **Step 1:** First, we need to provide the number of clusters, K, that need to be generated by this algorithm.
- **Step 2:** Next, choose K data points at random and assign each to a cluster. Briefly, categorize the data based on the number of data points.
- **Step 3:** The cluster centroids will now be computed.
- **Step 4:** Iterate the steps below until we find the ideal centroid, which is the assigning of data points to clusters that do not vary.
- 4.1 The sum of squared distances between data points and centroids would be calculated first.
- 4.2 At this point, we need to allocate each data point to the cluster that is closest to the others (centroid).
- 4.3 Finally, compute the centroids for the clusters by averaging all of the cluster's data points.

K-means implements the **Expectation-Maximization** strategy to solve the problem. The Expectation-step is used to assign data points to the nearest cluster, and the Maximization-step is used to compute the centroid of each cluster.

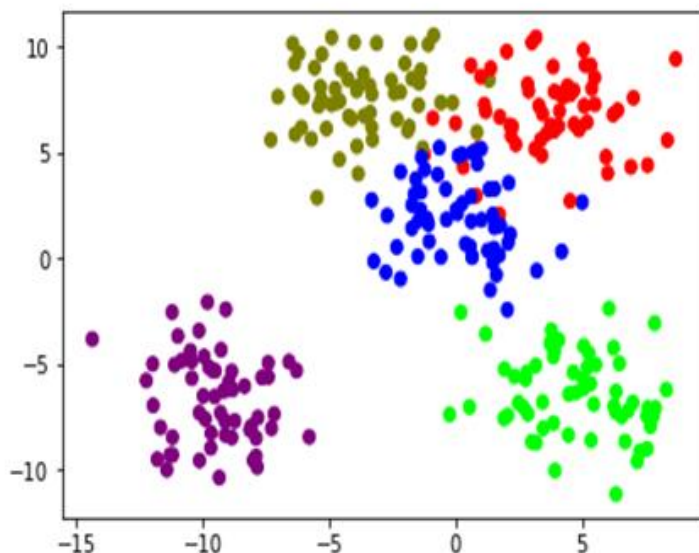


FIG 3.2 REPRESENTATION OF K-MEANS CLUSTERING ALGORITHM

3.9 WORKING

Since, machine learning model is built based on the car details given, it is required to create a form for Authentication in the Visual Studio. Authentication here comprises of the login details of the user. Once we enter the login details, we are directed to the next page that is Car details. Car details comprises of our car no, name etc.

Once we are done with the details of the car, we are being moved to the next page which is the car parking details. In this form, we are asked to fill the data and time of the car to be parked, after which we are given the slot number. Using all these data given by the user, a Machine Learning model is being built using the K-Means Clustering Algorithm. K-means clustering algorithm is an Algorithm which is effectively used for predictions and is very helpful in building various Machine Learning Models. This project is mainly proposed for two types of viewers. One is the customer or the enduser and the other is the administrator of the website. Some of the major functions of the product can be categorized under two different categories that are for the administrator and the user.

USER CHARACTERISTICS

There are two kinds of users for the Car parking allocation System. One is the customer and the other is the administrator. The customers do not need to have any prior training to use the application. The administrators would however need to be trained in order to use the application.

To describe more in detail, this project contains the main modules which comprises of Login Module

- Master Login Module
- Company Owner Login Module
- Admin Login Module
- New User Registration Module
- Adding Vehicle Details Module
- Car Parking Slot Module
- Car Allocation and Cost Estimation

Master Login:

This master login module is developed for passengers, where a user logging into his/her own account will view this panel. The operations included in this module are,

- 1) Admin view Data – Here entire user data is been accessed by the person who controls the parking in the system.
- 2) User view Data after Anonymization – Here user cannot see entire fields of data like money earning, cost per hour, car availability etc., Anonymization means part of the data to be viewed by the user.

Company Owner Login:

This module is only accessed by the owner or the proprietor of the parking space area. This is handled is particularly handled by him/her as it contains confidential details such as cost, initial cost etc. This is not dealt with any other person other than the owner himself. Hence this module is only accessed by him/her.

Admin Login:

Admin module is specifically developed for the people who have the rights to store the details of the car into the database. Here they will be storing the details of the car in order to get the car a slot in the parking slot section.

New User Registration:

When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. A person has to fill the details and submit. The details consist of firstname, lastname, password, Security Question, Security Answer, Mobile Number and city. This is done in order to get the vehicle registered.

When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. A person has to fill the details and submit. The details consist of firstname, lastname, password, Security Question, Security Answer,

Mobile Number and city. This is done in order to get the vehicle registered.

Adding Vehicle Details:

Once the user logged into the system, they can add the vehicles and its details. Such as,

- 1) Owner Name
- 2) Vehicle Number
- 3) Owner Driving license number
- 4) Vehicle model
- 5) Date and Time of Parking
- 6) Number of hours of Parking
- 7)Any Special Parking Credentials like offers etc.

These details are added for uploading in the database. These details will be stored and will be used for allocating the spaces with respect to the order it has been stored. These details are updated and stored by the admin, which is necessary for the successful allocation of the car parking space.

Car Parking Slot:

Based on the date and timing of the parking, an automatic allocation is done with the K-Means algorithm. In this module, a clear view of the cars allocated can be visualized with the text bounded on each location. A complete visualization of the car parking slots will be displayed on screen. This page provides a complete picturization in the grid format on all the parking allocation slots. A list of unallocated locations will be displayed. In addition, a clear picturization of the user views and their display data can be visualized in this module. This module provides a complete control of working in the system.

Car Allocation and Cost Estimation:

This module can be accessed from the user. It provides a clear picture on the past history of the vehicle. Any pending amounts related to the car and any extra charges which needs to be incorporated should be manipulated in this module. Any valet parking cum offers associated with the customer needs to be

manipulated will be added in this module. As a result, the cost estimations would be displayed here. The vehicles are also being updated in order to occupy the space in a proper manner. The K-Means Algorithm is implemented here and this helps in allocating the space in a systematic way. The details of the cars are also displayed on the side. There are also costs being assigned to each car to which users have to pay. This is entirely been done in Visual studio. Every icon is being designed using Visual Studio and the language used here is c#. Also, we are using SQL server as on our backend for the database connectivity.

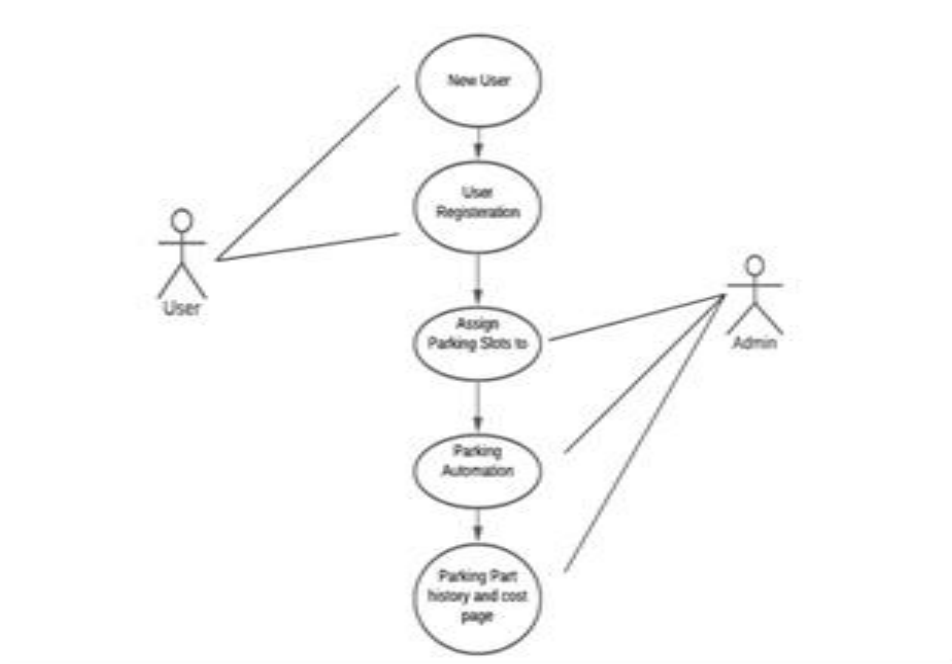
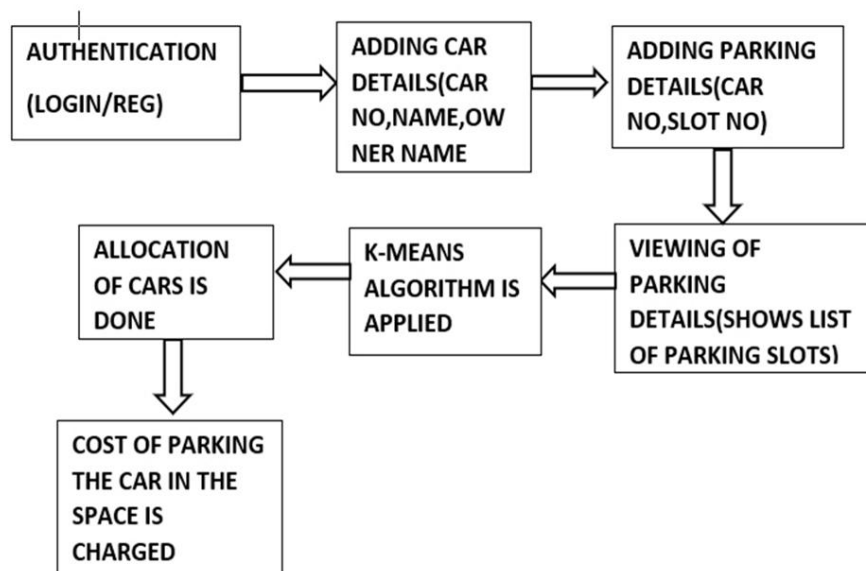


FIG 3.3 USE CASE DIAGRAM OF THE MODEL

3.10 PROJECT DESCRIPTION

The project aims to build a machine learning model using K-Means Clustering Algorithm which takes the data which is huge in number and groups them into clusters with similar data points. This project is helpful in predicting the car parking slots required for the car to be parked. This is not only helpful in parking but also allocates the car in a particular order. This project also focusses on reserved slots. This project mainly focusses on the K-Means Clustering Algorithm implemented as it plays an important role in predicting and allocating the space required for the car. It uses automation in the utmost manner.



3.4 WORKFLOW OF THE MODEL

CHAPTER 4

RESULTS AND DISCUSSION

4.1 SYSTEM REQUIREMENTS

- PROCESSOR: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
- RAM: 8GB
- SYSTEM TYPE: 64 bit Operating System
- STORAGE: 500 GB

4.2 RESULT

As a result, the main aim of this project was to make predictions based on the spaces given and make allocations for the car which is in a need to get the parking. Therefore it was evaluated by building a model using K-Means Clustering Algorithm which groups data into clusters and allocates them to the nearest data points given. This was achieved by using the language C#. This algorithm works well when compared to the rest of the algorithms used for building models. The K-Means Algorithm has an accuracy of 97%. This model is useful for people to use in an automotive way than in a manual way. It does better predictive analysis and allocations which cannot be achieved in a manual manner. In future, it can be developed into a more advanced model with better algorithm which has more efficiency and accuracy compared to our model.



FIG 4.1 RESULT

4.3 PERFORMANCE ANALYSIS

In this Analysis shown in fig 4.2, The representations of various algorithm is found in terms of its accuracy. Here, the accuracy is calculated by the sum of squared errors. K-Means is leading because it has 97% Accuracy in terms of performing and giving accurate results or Output. Also it is fast in processing data. K-Means is one of the most sought after algorithms in Machine Learning as it helps in grouping data in clusters and providing the desired outputs. It is used in prediction, allocation etc. It is an iterative algorithm that partitions the dataset into predefined distinct, non-overlapping subgroups which is otherwise called Clusters. And most importantly it is an unsupervised learning algorithm. Hence, it is efficient in this system being proposed here. The next Algorithm is FCFS which has an accuracy of 94% in terms of its performance. Here, the algorithm works on the process of First come First Serve basis. The purpose of using the algorithm is for allocating the car in a queue manner. To be more precise, this Algorithm is used for Queue allocation of Cars. Since the working of Queues is based on First Come First Serve basis, As an algorithm it helps in allocating the cars. But the drawback of this algorithm is it takes time in allocating the cars as it has to find the spaces and allocates the car one by one. As a result it is good in performing and yielding accurate results but it lags when compared to K-Means Algorithm. To make an

analysis, it is one of the second most performing algorithms after K-Means. It just makes a 3% difference from K-Means Algorithm. And the last one is Dijkstra Algorithm which has an accuracy of 90%. Here Dijkstra Algorithm helps in finding the shortest path. In a scenario like parking allocation this algorithm finds the shortest distance of parking space and allocates the space in a proper manner. Under this it takes into consideration many factors like Geographic Information System (GIS), Global positioning System (GPS), traffic information system, routing system and many more. This also has another function calculating the distance of the path and making spaces for cars to be allocated in. Here each parking slot is scanned and as a result, nodes are generated in order to reach the destination. This algorithm is known for its best way of finding the shortest path. But since there are a lot of calculations involved in this process, it is quite uncertain to find all the shortest paths if given an empty space.

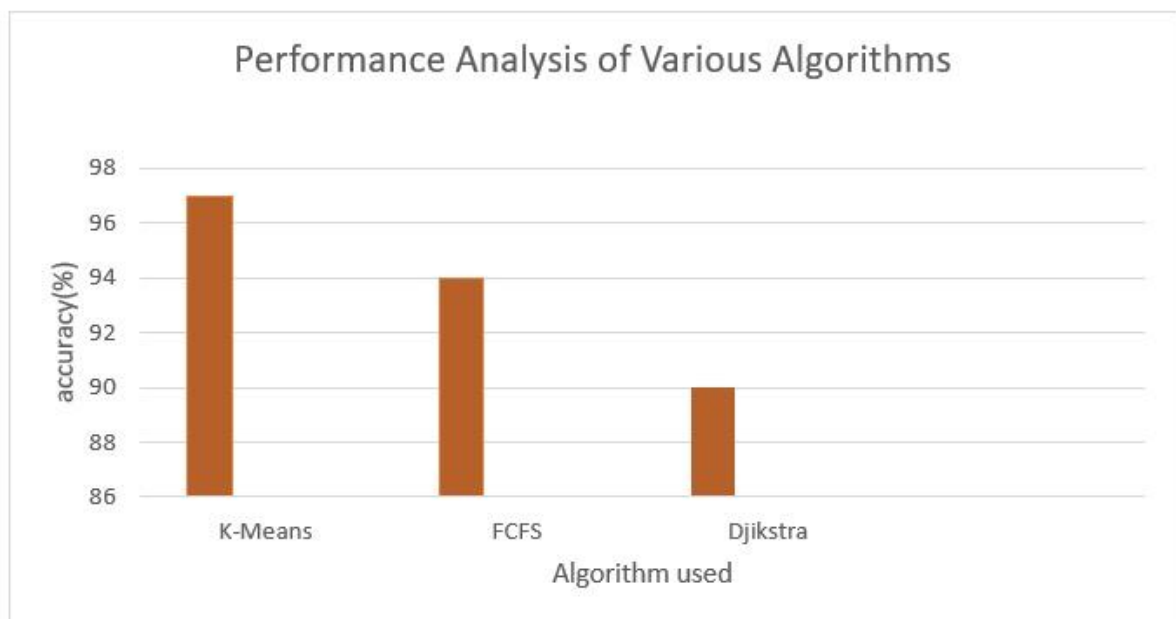


FIG 4.2 PERFORMANCE ANALYSIS OF VARIOUS ALGORITHMS

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This project is undertaken using machine learning and evaluates the performance of prediction by using K-Means Clustering Algorithm. In our proposed model, it is built using the above algorithm. The implementation of above system would help in better prediction of whether the parking slot is available or not and does a better job by allocating the spaces to the car after collecting the details of the car. This gives results more accurate than by assumptions or manual methods.

5.2 FUTURE WORK

This project describes prediction of spaces or slots required for the car to be parked and also allocates the car to the particular space by using the K-Means Clustering Algorithm. In future, the parking can be done in an automative way like the way it is implemented in this project by making the best uses of Machine Learning. By doing so, it will be helpful for getting the results in an accurate manner unlike the normal or manual method.

5.3 REFERENCES

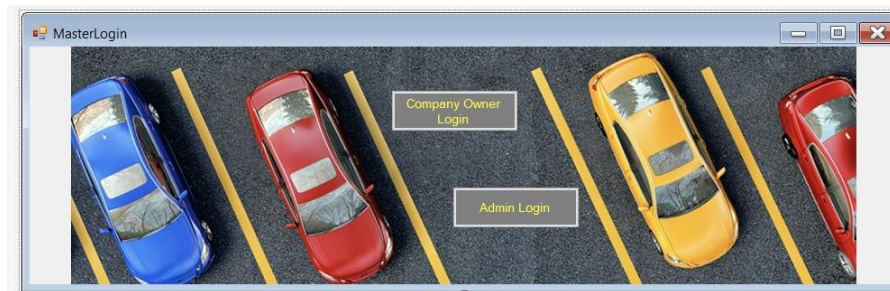
1. Nyambal, J., & Klein, R. (2017, November). Automated parking space detection using convolutional neural networks. In 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech) (pp. 1-6). IEEE.
2. Gupta, R. K., & Rani, G. (2020, March). Machine Learning and IoT based Real Time Parking System: Challenges and Implementation. In Proceedings of the International Conference on Innovative Computing & Communications (ICICC).
3. Siddiqui, S. Y., Khan, M. A., Abbas, S., & Khan, F. (2020). Smart occupancy detection for road traffic parking using deep extreme learning machine. *Journal of King Saud University-Computer and Information Sciences*.
4. Ali, G., Ali, T., Irfan, M., Draz, U., Sohail, M., Glowacz, A., ... & Martis, C. (2020). IoT based smart parking system using deep long short memory network. *Electronics*, 9(10), 1696.

5. Srivastava N., Hinton G., Krizhevsky A., Sutskever I. and Salakhutdinov R., Dropout: A Simple way to prevent Neural Networks from overfitting, Journal of Machine Learning Research, Springer, 2017.
6. H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: path planning and control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp.396–410, 2015.
7. F.Bock, S.Di Martino, and M. Sester, "Data-driven approaches for smartparking," in Machine Learning and Knowledge Discovery in Databases.Springer International Publishing, 2017, pp. 358–362
8. Shree, B. (2017). Design and Implementation of Automated Car Parking System using RFID. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6(2), ISSN: 2319 5940
9. P. Zips, M. Böck, and A. Kugi, "Optimisation based path planning *Robotics and Autonomous Systems*, vol. 79, pp. 1–11, 2015.
- 10.Z. Mahmood, "Towards a fully automated car parking system," IETIntelligent Transport Systems, vol. 13, pp. 293–302(9), February 2019.
- 11.Bevis Jinila, K.Komathy (2015). Rough set based fuzzy scheme for clustering and cluster head selection in VANET. Elektronika Ir Elektrotechnika, 21(1), 54-59.
- 12.Jinila, B. (2015). Cluster oriented ID based multi-signature scheme for traffic congestion warning in Vehicular Ad hoc Networks. In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2 (pp. 337-345). Springer, Cham.

APPENDIX

A.SCREENSHOTS

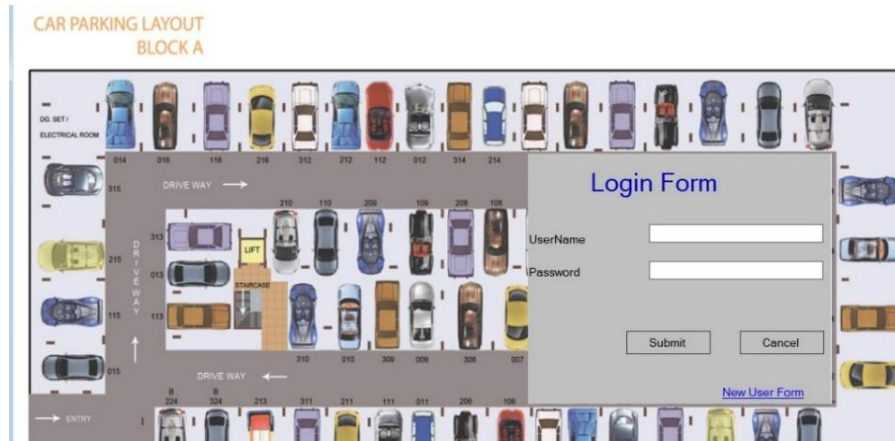
1. Master Login



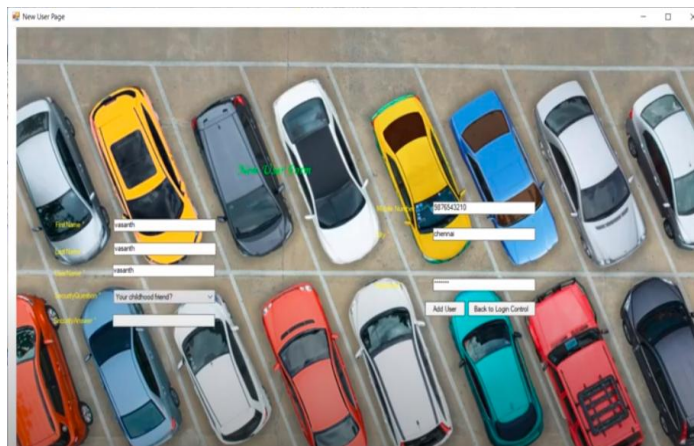
2. Company Owner Login



3. AdminLogin



4. New User Form



5. Adding Vehicle Details

Add Vehicle Details

Owner Name

Car Number

Car Model

Parking Date Fetch Time

Parking Time

Cost Per Hour

Initial Advance

6. Car Parking Slot

7. Car Allocation and Cost Estimation

ProvideRating

Car Number *

Car Allocation and Cost Estimation

id	ownername	camumber	camodel	Parkingdate	parkingtime	costPerhour
1	venkatesan prabu	TN45 3456	BMW	29-03-2021 19:1...	19:17	5

Parking Date: Current Date: Fetch Time

Parking Time: Current Time:



B. SOURCE CODE

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace Library
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MasterLogin());
        }
    }
}
```

K-Means.cs

```
using System;

using System.Collections;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Reflection;

using System.Text;

namespace Library
{
    /// <summary>
```

```

/// Defines a property or field as an attribute to use for the k-means clustering
/// </summary>
[AttributeUsage(AttributeTargets.Property)]
public sealed class KMeansValueAttribute : Attribute{}

public delegate double KMeansCalculateDistanceDelegate(double[] point, double[]
centroid);

/// <summary>
/// Provides a simple implementation of the k-Means algorithm. This solution is quite
simple and does not support any parallel execution as of yet.
/// </summary>

public static class KMeans
{
    private static double[][] ConvertEntities<T>(T[] items)
    {
        var type = typeof (T);
        var data = new List<double[]>();

        // If the type is an array type
        if (type.IsArray&&type.IsAssignableFrom(typeof(double[])))
        {
            foreach (var item in items)
            {
                var val = item as double[];
                data.Add(val);
            }
        }

        return data.ToArray();
    }
}

```

```
}
```

```
var getters = new List<MethodInfo>();
```

```
// Iterate over the type and extract all the properties that have the  
KMeansValueAttribute set and use them as attributes
```

```
var attribType = typeof (KMeansValueAttribute);
```

```
foreach (var property in  
type.GetProperties(BindingFlags.Instance|BindingFlags.Public))
```

```
{
```

```
var attribs = property.GetCustomAttributes(attribType,  
false).OfType<KMeansValueAttribute>().ToArray();
```

```
if( attribs.Length<= 0 )
```

```
continue;
```

```
var getter = property.GetGetMethod();
```

```
if( getter == null )
```

```
throw new InvalidOperationException("No public getter for property  
"+property.Name+"". All properties marked with the KMeansValueAttribute must have  
a public getter");
```

```
if( !property.PropertyType.IsAssignableFrom(typeof(double)) &&
```

```
!property.PropertyType.IsAssignableFrom(typeof(int)) &&
```

```
!property.PropertyType.IsAssignableFrom(typeof(float)) &&
```

```
!property.PropertyType.IsAssignableFrom(typeof(long)) &&
```

```
!property.PropertyType.IsAssignableFrom(typeof(decimal)) &&
```

```
!property.PropertyType.IsAssignableFrom(typeof(short)) )
```

```

throw new InvalidOperationException("Property type
"+property.PropertyType.Name+" for property "+property.Name+" cannot be
assigned to System.Double. ");

getters.Add(getter);
}

foreach (var item in items)
{
    List<double> values = new List<double>(getters.Count);
    foreach (var getter in getters)
        values.Add(Convert.ToDouble(getter.Invoke(item, null)));
    data.Add(values.ToArray());
}

return data.ToArray();
}

public static KMeansResults<T> Cluster<T>(T[] items, int clusterCount, int
maxIterations, KMeansCalculateDistanceDelegate calculateDistanceFunction = null,
int randomSeed = 0, int[] initialCentroidIndices = null)
{
    double[][] data = ConvertEntities(items);

    if (calculateDistanceFunction == null)
        calculateDistanceFunction = CalculateDistance;

    bool hasChanges = true;
    int iteration = 0;
    double totalDistance = 0;
    int numData = data.Length;

```

```

int numAttributes = data[0].Length;

// Create a random initial clustering assignment
int[] clustering = InitializeClustering(numData, clusterCount, randomSeed);

// Create cluster means and centroids
double[][] means = CreateMatrix(clusterCount, numAttributes);
int[] centroidIdx = new int[clusterCount];
int[] clusterItemCount = new int[clusterCount];

// If we specify initial centroid indices then let's assign clustering based on those
immediately
if (initialCentroidIndices != null && initialCentroidIndices.Length == clusterCount)
{
    centroidIdx = initialCentroidIndices;

    AssignClustering(data, clustering, centroidIdx, clusterCount,
        calculateDistanceFunction);

    //          Debug.WriteLine("Pre-Seeded Centroids resulted in initial clustering: " +
    string.Join(",", clustering.Select(x => x.ToString()).ToArray()));
} // Perform the clustering

while (hasChanges && iteration < maxIterations)
{
    clusterItemCount = new int[clusterCount];

    totalDistance = CalculateClusteringInformation(data, clustering, ref means, ref
        centroidIdx, clusterCount, ref clusterItemCount, calculateDistanceFunction);
}

```

```

//          Debug.WriteLine("----- Iter: " + iteration);

//          Debug.WriteLine("Clustering: " + string.Join(",", clustering.Select(x
=>x.ToString()).ToArray()));

//          Debug.WriteLine("Means: " + string.Join(",", means.Select(x => "[" +
string.Join(",", x.Select(y =>y.ToString("#0.0")).ToArray()) + "]").ToArray()));

//          Debug.WriteLine("Centroids: " + string.Join(",", centroidIdx.Select(x
=>x.ToString()).ToArray()));

//          Debug.WriteLine("Cluster Counts: " + string.Join(",",
clusterItemCount.Select(x =>x.ToString()).ToArray()));

hasChanges = AssignClustering(data, clustering, centroidIdx, clusterCount,
calculateDistanceFunction);

++iteration;

}

// Create the final clusters

T[][] clusters = new T[clusterCount][];
for (int k = 0; k < clusters.Length; k++)
clusters[k] = new T[clusterItemCount[k]];
int[] clustersCurIdx = new int[clusterCount];
for (int i = 0; i < clustering.Length; i++)
{
clusters[clustering[i]][clustersCurIdx[clustering[i]]] = items[i];
++clustersCurIdx[clustering[i]]
}

// Return the results

return new KMeansResults<T>(clusters, means, centroidIdx, totalDistance);

```

```

}

private static int[] InitializeClustering(int numData, int clusterCount, int seed)
{
    var rnd = new Random(seed);
    var clustering = new int[numData];

    for (int i = 0; i < numData; ++i)
        clustering[i] = rnd.Next(0, clusterCount);

    return clustering;
}

private static double[][] CreateMatrix(int rows, int columns)
{
    var matrix = new double[rows][];

    for (int i = 0; i < matrix.Length; i++)
        matrix[i] = new double[columns];

    return matrix;
}

private static double CalculateClusteringInformation(double[][] data, int[] clustering, ref
double[][] means, ref int[] centroidIdx,

int clusterCount, ref int[] clusterItemCount,
KMeansCalculateDistanceDelegate calculateDistanceFunction)
{
    // Reset the means to zero for all clusters

```

```

foreach (var mean in means)
for (int i = 0; i<mean.Length; i++)
mean[i] = 0;

// Calculate the means for each cluster

// Do this in two phases, first sum them all up and then divide by the count in each
cluster
for (int i = 0; i<data.Length; i++)
{
// Sum up the means
var row = data[i];
var clusterIdx = clustering[i]; // What cluster is data i assigned to
++clusterItemCount[clusterIdx]; // Increment the count of the cluster that row i is
assigned to
for (int j = 0; j <row.Length; j++)
means[clusterIdx][j] += row[j];
}

// Now divide to get the average
for (int k = 0; k <means.Length; k++)
{
for (int a = 0; a < means[k].Length; a++)
{
int itemCount = clusterItemCount[k];
means[k][a] /= itemCount>0 ?itemCount : 1;
}
}
}

```

```

double totalDistance = 0;

// Calc the centroids

double[] minDistances = new double[clusterCount].Select(x
=>double.MaxValue).ToArray();

for (int i = 0; i<data.Length; i++)
{
    var clusterIdx = clustering[i]; // What cluster is data i assigned to
    //var distance = CalculateDistance(data[i], means[clusterIdx]);
    var distance = calculateDistanceFunction(data[i], means[clusterIdx]);
    totalDistance += distance;
    if (distance < minDistances[clusterIdx])
    {
        minDistances[clusterIdx] = distance;
        centroidIdx[clusterIdx] = i;
    }
}

//double totalCentroidDistance = minDistances.Sum();

return totalDistance;
}

private static bool AssignClustering(double[][] data, int[] clustering, int[] centroidIdx, int
clusterCount, KMeansCalculateDistanceDelegate calculateDistanceFunction)
{
    bool changed = false;

    for (int i = 0; i<data.Length; i++)
    {

```

```

double minDistance = double.MaxValue;
int minClusterIndex = -1;
for (int k = 0; k < clusterCount; k++)
{
    double distance = calculateDistanceFunction(data[i], data[centroidIdx[k]]);
    if (distance < minDistance)
    {
        minDistance = distance;
        minClusterIndex = k;
    }

    // todo: track outliers here as well and maintain an average and std calculation for the
    distances!
}

// Re-arrange the clustering for datapoint if needed
if (minClusterIndex != -1 && clustering[i] != minClusterIndex)
{
    changed = true;
    clustering[i] = minClusterIndex;
}
}return changed;
}

private static double CalculateDistance(double[] point, double[] centroid)
{

```

```

// For each attribute calculate the squared difference between the centroid and the
point

double sum = 0;

for (int i = 0; i<point.Length; i++)

sum += Math.Pow(centroid[i] - point[i], 2);

return Math.Sqrt(sum);

//return Math.Sqrt(point.Select((t, i) =>Math.Pow(centroid[i] - t, 2)).Sum()); // LINQ is
slower than doing the for-loop!

}

}

public class KMeansResults<T>

{

public T[][] Clusters { get; private set; }

public double[][] Means { get; private set; }

public int[] Centroids { get; private set; }

public double TotalDistance{ get; private set; }

public KMeansResults(T[][] clusters, double[][] means, int[] centroids, double
totalDistance)

{

Clusters = clusters;

Means = means;

Centroids = centroids;

TotalDistance = totalDistance;

}

}

}

```

Master Login.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Library
{
    public partial class MasterLogin : Form
    {
        public MasterLogin()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Login login = new Login();
            login.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            UserLogin login = new UserLogin();
            login.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
        }

        private void MasterLogin_Load(object sender, EventArgs e)
        {
        }
    }
}
```

User Login.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Library
{
    public partial class UserLogin : Form
    {
        public UserLogin()
        {
            InitializeComponent();
        }

        private void LoginSuccess(object sender, System.EventArgs e)
        {
        }

        private void LoginSuccess1(object sender, System.EventArgs e)
        {
            AddInputscsAddInputscs = new AddInputscs();
            AddInputscs.Show();
            this.Hide();
        }

        private void NewUserFormControlTransfer(object sender, System.EventArgs e)
        {
            NewUserForm NUF = new NewUserForm();
            NUF.Show();
            this.Hide();
        }

        private void UserLogin_Load(object sender, EventArgs e)
        {
        }

        private void loginControl1_Load(object sender, EventArgs e)
```

```
{  
}  
}  
}
```

Admin Login.cs

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
using Library;  
  
namespace Library  
{  
    public partial class Login : Form  
    {  
        public Login()  
        {  
            InitializeComponent();  
            ObjectAccess.ConnectDBdbO = new ObjectAccess.ConnectDB();  
            private void NewUserFormControlTransfer(object sender, System.EventArgs e)  
            {  
                NewUserForm NUF = new NewUserForm();  
                NUF.Show();  
                this.Hide();  
            }  
  
            private void Login_Load(object sender, EventArgs e)  
            {  
  
            }  
  
            private void manufacturerLogin1_LoginSuccess1(object sender, System.EventArgs e)  
            {  
  
                ViewData VD = new ViewData();  
                VD.Show(); this.Hide();  
            }  
        }  
    }  
}
```

```
private void LoginSuccess(object sender, System.EventArgs e)
{

}
```

```
private void LoginSuccess1(object sender, System.EventArgs e)
{
    CostFinal VD = new CostFinal();
    VD.Show(); this.Hide();
}
```

```
private void loginControl1_Load(object sender, EventArgs e)
{

}
```

```
private void btnSubmit_Click(object sender, EventArgs e)
{
    if (txtUserName.Text == "Admin" &&txtPwd.Text == "Admin")
    {
        ViewDataViewData = new ViewData();
        ViewData.Show();
        this.Hide();
    }
    else
        Console.WriteLine("Please give valid info");
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
```

```
private void loginControl2_Load(object sender, EventArgs e)
{

}
```

```
private void label5_Click(object sender, EventArgs e)
{

}
```

```
private void label1_Click(object sender, EventArgs e)
{
```

```
}  
}  
}
```

New User Form.cs

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
  
namespace Library  
{  
    public partial class NewUserForm : Form  
    {  
        public NewUserForm()  
        {  
            InitializeComponent();  
        }  
  
        private void NewUserForm_Load(object sender, EventArgs e)  
        {  
  
        }  
        private void Login(object sender, System.EventArgs e)  
        {  
            Login login = new Login();  
            login.Show();  
            this.Hide();  
        }  
        private void NewUserSuccess(object sender, System.EventArgs e)  
        {  
            Login login = new Login();  
            login.Show();  
            this.Hide();  
        }  
  
        private void newUserControl1_Load(object sender, EventArgs e)  
        {  

```

```
}  
  
}  
}
```

Adding Vehicle Details.cs

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
using ObjectAccess;  
using System.Globalization;  
namespace Library  
{  
    public partial class AddInputscs : Form  
    {  
        public AddInputscs()  
        {  
            InitializeComponent();  
        }  
        ObjectAccess.ConnectDBdbObject = new ConnectDB();  
  
        private void button1_Click(object sender, EventArgs e)  
        {  
            ViewData VD = new ViewData();  
            VD.Show(); this.Hide();  
        }  
  
        private void btnSubmit_Click(object sender, EventArgs e)  
        {  
  
        }  
  
        private void AddInputscs_Load(object sender, EventArgs e)  
        {  
  
        }  
  
        private void button3_Click(object sender, EventArgs e)  
        {
```

```

int val = dbObject.insertcar(txt_ownerName.Text,
txt_CarNumber.Text, txt_CarModel.Text, textBox1.Text,
txt_ParkingTime.Text, txt_Duration.Text ,
txt_advance.Text

);
if (val == 1)
{
    MessageBox.Show("Car Details Added Successfully !!!");

}
else
{
    MessageBox.Show("Issue in the connectivity");
}
}

private void button2_Click(object sender, EventArgs e)
{
    CarParkingUpdationCarParkingUpdation = new CarParkingUpdation();
    CarParkingUpdation.Show();
    this.Hide();
}

private void button1_Click_1(object sender, EventArgs e)
{
    AddInputscsAddInputscs = new AddInputscs();
    AddInputscs.Show();
    this.Hide();

}

private void button5_Click(object sender, EventArgs e)
{
    CostFinalCostFinal = new CostFinal();
    CostFinal.Show();
    this.Hide();
}

private void button4_Click(object sender, EventArgs e)
{
    CarParkingUpdationCarParkingUpdation = new CarParkingUpdation();
    CarParkingUpdation.Show();
    this.Hide();
}

```

```

}

private void button6_Click(object sender, EventArgs e)
{
    textBox1.Text = DateTime.Now.ToString();
    DateTime dt = DateTime.Parse(DateTime.Now.ToString());

    txt_ParkingTime .Text = dt.ToString("HH:mm");

}

private void txt_ownerName_TextChanged(object sender, EventArgs e)
{
}
}
}
}

```

Car Parking Updation.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.Mail;
using System.IO;
using System.IO;
using System.Threading.Tasks;
using System.Data.SqlClient;

namespace Library
{
    public partial class CarParkingUpdation : Form
    {
        public CarParkingUpdation()
        {
            InitializeComponent();
        }
    }
}

```

```

ObjectAccess.ConnectDB      _EngleWood__dbo_C_198_d      =      new
ObjectAccess.ConnectDB();
GoogleSpatialManipulation.GoogleSpatialManipulationGoogleSpatialManipulation  =
new GoogleSpatialManipulation.GoogleSpatialManipulation();

private void CarParkingUpdation_Load(object sender, EventArgs e)
{

foreach(DataRowdr in _EngleWood__dbo_C_198_d.FetchCarNumber().Rows)

{
cmb_vehcleNumber.Items.Add(dr[0].ToString());

}

}

private void button9_Click(object sender, EventArgs e)
{
DataTableGetAllocated      =
_EngleWood__dbo_C_198_d.ParkingAllocation(cmb_vehcleNumber.SelectedItem.
ToString()
);

foreach (DataRowdr in GetAllocated.Rows)
{
Button controleText = (Button)groupBox1.Controls[dr[3].ToString()];
controleText.Text = dr[0].ToString() + " \n " + dr[1].ToString() + " \n \n " +
dr[2].ToString();
}

try
{
dataGridView1.DataSource      =
_EngleWood__dbo_C_198_d.GetAllocated1(cmb_vehcleNumber.SelectedItem.ToS
tring());
}

catch (Exception ex)
{
dataGridView1.DataSource = _EngleWood__dbo_C_198_d.GetAllocated2();
}
}

```

```

private void button36_Click(object sender, EventArgs e)
{
    try
    {
        DataTable GetAllocated = _EngleWood__dbo_C_198_d.GetAllocated();

        foreach (DataRow dr in GetAllocated.Rows)
        {
            Button controleText = (Button)groupBox1.Controls[dr[3].ToString()];
            controleText.Text = dr[0].ToString() + " \n " + dr[1].ToString() + " \n \n " +
            dr[2].ToString();
        }
    }
    try
    {
        dataGridView1.DataSource =
        _EngleWood__dbo_C_198_d.GetAllocated1(cmb_vehicleNumber.SelectedItem.ToS
tring());
    }

    catch (Exception ex)
    {
        dataGridView1.DataSource = _EngleWood__dbo_C_198_d.GetAllocated2();
    }
}
catch (Exception ex)
{
}
}

private void button49_Click(object sender, EventArgs e)
{
    try
    {
    }
    catch (Exception ex) { }
}

private void cmb_vehicleNumber_SelectedIndexChanged(object sender, EventArgs
e)
{
    txt_Date.Text =
    _EngleWood__dbo_C_198_d.FetchParkingDate(cmb_vehicleNumber.SelectedItem.
ToString()).Rows[0][0].ToString();
}

```

```

txt_time.Text =
EngleWood__dbo_C_198_d.FetchParkingDate(cmb_vehicleNumber.SelectedItem.
ToString()).Rows[0][1].ToString();

}

private void button8_Click_1(object sender, EventArgs e)
{
this.Close();
}

private void button50_Click(object sender, EventArgs e)
{
AddInputscsAddInputscs = new AddInputscs();
AddInputscs.Show();
this.Hide();
}

private void button49_Click_1(object sender, EventArgs e)
{
CarParkingUpdationCarParkingUpdation = new CarParkingUpdation();
CarParkingUpdation.Show();
this.Hide();
}

private void button48_Click(object sender, EventArgs e)
{
CostFinalCostFinal = new CostFinal();
CostFinal.Show();
this.Hide();
}

private void button51_Click(object sender, EventArgs e)
{

SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-
JC4U74BL;Integrated Security=true;InitialCatalog=Carparking");

conn.Open();

DataSet ds = new DataSet();
SqlDataAdapter adapter = new SqlDataAdapter();

```

```

SqlCommandcmd = new SqlCommand("select carnumber , carmodel FROM
[Carparking].[dbo].[cardetails]", conn);

cmd.CommandType = CommandType.Text;

adapter.SelectCommand = cmd;

adapter.Fill(ds);
conn.Close();

StreamWriter file = new StreamWriter(@"D:\Final Year Project\Parking Allocation
System\KMeans\bin\Debug\WriteLines2.txt");
file.Flush();
foreach (DataRowdr in ds.Tables[0].Rows)
{

file.Write(dr[0].ToString() + " => " + dr[1].ToString()+"\n");
}
file.Close();

MessageBox.Show("Machine learning algorithm is implemented here");

}
}
}

```

CostFinal.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using ObjectAccess;
using System.Data.SqlClient;
using System.Globalization;
namespace Library
{
public partial class CostFinal : Form
{

```

```

public CostFinal()
{
    InitializeComponent();
}
ObjectAccess.ConnectDB      _EngleWood__dbo_C_198_d      =      new
ObjectAccess.ConnectDB();
ObjectAccess.ConnectDBobj = new ConnectDB();
private void ProvideRating_Load(object sender, EventArgs e)
{
    try
    {
        foreach (DataRow dr in _EngleWood__dbo_C_198_d.FetchCarNumber1().Rows)
        {
            cmbProductName.Items.Add(dr[0].ToString());
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("No Items available to provide rating");
    }
    cmbProductName.SelectedIndex = 0;
}

```

```

private void button2_Click_1(object sender, EventArgs e)
{
    AddInputscsAddInputscs = new AddInputscs();
    AddInputscs.Show();
    this.Hide();
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    CarParkingUpdationCarParkingUpdation = new CarParkingUpdation();
    CarParkingUpdation.Show();
    this.Hide();
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    CostFinalCostFinal = new CostFinal();
    CostFinal.Show();
}

```

```

this.Hide();
}

private void cmbProductName_SelectedIndexChanged(object sender, EventArgs e)
{

    try
    {
        dataGridView1.DataSource =
        _EngleWood__dbo_C_198_d.GetAllocated1(cmbProductName.SelectedItem.ToString());
        txt_Parkingdate.Text =
        _EngleWood__dbo_C_198_d.GetAllocated1(cmbProductName.SelectedItem.ToString()).Rows[0][4].ToString();

        DateTime dt = DateTime.Parse(txt_Parkingdate.Text);

        txt_Parking__time.Text = dt.ToString("HH.mm");

        txt_currentDate.Text = DateTime.Now.ToString();

    }

    catch (Exception ex)
    {
        dataGridView1.DataSource = _EngleWood__dbo_C_198_d.GetAllocated2();
    }
}

float diff = 0; float value = 0;
private void button6_Click(object sender, EventArgs e)
{
    txt_currentDate.Text = DateTime.Now.ToString();
    DateTime dt = DateTime.Parse(DateTime.Now.ToString());

    txt_currentTime.Text = dt.ToString("HH.mm");
    float flt1 = float.Parse(txt_currentTime.Text);
    float flt2 = float.Parse(txt_Parking__time.Text);

    if (flt1 > flt2)
    {
        diff = flt1 - flt2;
        diff = Convert.ToInt32(diff);
    }
}

```

```

else
{
diff = flt1+12 - flt2;
diff = Convert.ToInt32(diff);
}

MessageBox.Show("The duration of your vehicle is" +diff.ToString());

int costperhour = Convert.ToInt32(
_EngleWood__dbo_C_198_d.FetchCostperhour(cmbProductName.SelectedItem.ToS
tring()).Rows[0][0].ToString());

value = diff * costperhour;
if (value == 0)
{
value = 5;
}
MessageBox.Show("Parking charge for your vehicle is " + value);

}

private void btnSubmit_Click(object sender, EventArgs e)
{
int val =
_EngleWood__dbo_C_198_d.updatecost(cmbProductName.SelectedItem.ToString(),
value.ToString());

_EngleWood__dbo_C_198_d.freeup(cmbProductName.SelectedItem.ToString())
;

MessageBox.Show("The cost of the vehicle is updated successfully");

}

private void button1_Click(object sender, EventArgs e)
{

}

private void label3_Click(object sender, EventArgs e)
{

```

}
}
}
}

C.PUBLICATION WITH PLAGIARISM REPORT

Research Paper Submitted For IEEE Publication.

Smrithii_Nithya_19-03- 2022_IAIM 2022.doc

by Smrithii_nithya_19-03-2022_iaim 2022.doc Smrithii_nithya_19-03-2022_iaim 2022.doc

Submission date: 19-Mar-2022 02:21AM (UTC-0400)

Submission ID: 1787647142

File name: Smrithii_Nithya_19-03-2022_IAIM_2022.doc (1.61M)

Word count: 3550

Character count: 1805

An Efficient Parking Allocation Scheme based on K-Means Clustering Method

Smrithi.R, Nithya.B, Y. Bevis Jinila, S. Prayla Shyry, R.Subhashini
School of Computing,
Sathyabama Institute of Science and Technology, Chennai, India.
Corresponding Author: ybevis@gmail.com

Abstract—With the development of the automobile sector in our country over the years now, the vehicles in use have constantly grown to new heights. As a result, when considering the rate of expansion in the number of motorized vehicles, it greatly outpaces the parking rate facility building. The idea of ongoing amelioration and development of parking distribution systems has always been important to the city's smooth operation. As a result, building an effective and dynamic parking algorithm will be a significant step forward in addressing the problem of urban parking shortages. As a result, utilizing the K-Means Algorithm, this study developed a strategy for parking allocation. The findings of the experiments reveal that a parking prediction and allocation model based on K-Means Clustering can not only anticipate and allocate parking spaces in real time, but it also has the ability to enhance the rate of utilization of different types of parking slots. As a result, it makes a substantial contribution to smarter and more sustainable urban parking management. In this paper an efficient parking allocation model is proposed and the research findings to predict the number of available parking spaces is discussed.

Keywords—Dynamic parking model, K-Means clustering, parking allocation, smart cities.

I. INTRODUCTION (HEADING 1)

The amount of private cars possessed by citizens of the country has been expanding year after year, owing to the country's economic progress and the increased manufacturing of motor vehicles in the automobile sector. As a result, parking-related issues have become more prevalent, as finding parking places has become difficult due to a scarcity of available parking spaces around the city, particularly in the city's central districts. As a result, drivers spend more time looking for parking spots, causing traffic congestion, pollution, and increased fuel consumption in the city. People attempting to find a parking spot account for around 30% of traffic congestion in urban areas. As a result, finding a parking spot in a city has always been a struggle for residents. Throughout 25% - 30% of automobiles in urban areas of major cities around the world would take an average of 8 minutes to search for a proper parking space.

The existing models, on the other hand, are of no help to persons who drive automobiles because they do not address the necessity for finding a vacant spot and allocating the car. The algorithms of Deep learning and Machine Learning have evolved efficiently, introducing a new method for analyzing traffic on a bigger scale. Parking models which prove to be dynamic, amalgamate parking demand prediction and allocation approaches to produce greater forecast results. As a matter of fact it instigates faster reactions, which marks appropriate for the traffic environment currently taking place in our country. The approach of sharing parking lots is a practical method of utilizing parking spots and resolving

parking issues, particularly in downtown areas. However, owing to the reason of the performance of the parking lot sharing, the approach has been controlled by demand and availability of parking, and because of this there is a less room for it to happen. As a result, as a solution to this recurring problem, this paper introduces a parking model which is much dynamic based on Machine Learning which helps in predicting and allocating the spaces for Automobiles. This also establishes the strategy for when to employ parking lot sharing. As a result, improved car parking assignment may be possible.

II. LITERATURE REVIEW

This section featured here analyses the merits and demerits of the works done by various authors which have been done prior and is related to the proposed work presented here. Finding a parking spot has become an issue that should not be overlooked; it takes time and effort. Using data from the University of the Witwatersrand, we employed methods for computer vision to determine the way of the parking slot implemented. This paper shows how to use Caffe and the Nvidia DiGITS framework to create a parking slot set up on real time categorization system based on Convolutional Neural Networks (CNN) techniques [1].

Ra. Kumar Gupta and Geetha Rani et al. (2020) [2] titled 'Machine Learning and IoT based Real time parking system: Challenges and Implementation'. They have done a work on a car parking system using Machine Learning and IOT describing its Challenges and the way it is implemented. The methodology examined in this paper has accomplished the objective to develop a model which will predict the particular space for the car to be parked. The machine learning model used here helps us to easily understand the way it implements. This will help the Driver to park his/her car in an efficient manner. But considering the demerits into account, it is understood that the system does not help the Drivers to understand whether the given area is reserved or unreserved. Hence, there needs to be a way to locate the area as reserved or unreserved by any means of identification like giving some color which will indicate if the car parking space is available or not.

Making Predictions of the locations of parking slots or spaces is a never-ending issue that is crucial to our daily lives. Artificial neural networks are employed in this paper to forecast the parking position, which will assist drivers in deciding on a safe area to halt. As a result of this strategy, traffic on the highways becomes more familiar and comfortable, resulting in less turbulence. The measures of reliability is accomplished with an error rate considered to be marginal, utilizing the Deep Extreme Learning Machine (DELm) technique [3], minimizing the skeptical propensity. Here, deep learning neural networks are used instead of

feedforward and backward propagation. The study emphasizes the Proposed Car Parking Space Prediction (CPSP) which gives a detailed overview on the challenges of finding parking slots.

In this system there is a concept of Deep learning being used. The study of Deep learning is a type of advanced learning which has evolved from Machine Learning and artificial Intelligence which gains knowledge of data the way humans do. Here, it makes use of neural networks which involves a lot of layers.

In this paper, Convolutional Neural Network (CNN) [4] and various other machine learning algorithms are being used which yields good results. The idea behind the paper is it helps to locate the free space by identifying its color and with the help of convolutional layers being created. And this is considered to be the merit of this work. But when it comes to its demerits, it takes a long time in processing the input data given to it. That is because, deep learning is a study that takes input data for processing and would take huge amount of time in delivering the result. Improved parking ability using various methodologies is discussed in [5-10]. The schemes on Intelligent Transportation Systems have inculcated various other sub domains like smart parking schemes to be employed to establish an efficient scheme. The schemes on smart cities and its deployment methods are discussed in [11-14].

III. PROPOSED WORK

The work proposed here is a system being built using K-Means Algorithm which will help the users to predict and allocate if the car can be parked in the allocated space or not. This is more effective in finding car spaces in an automated way more than that of a manual way. Here, a degenerative technique is used for finding the free space and allocating based on the time slots and locations is done on this system.

A. Architecture of the Proposed Model

Here, the system architecture shown in fig 1 represents the cars being parked on the given time slot. The first process is the machine learning algorithm called K-Means Algorithm is used for building the model. This is the algorithm which will take the input data and convert them into clusters. The next step is to build an efficient model which in order will predict the space according to the availability and helps the car to be allocated in a given space according to the clusters being obtained.

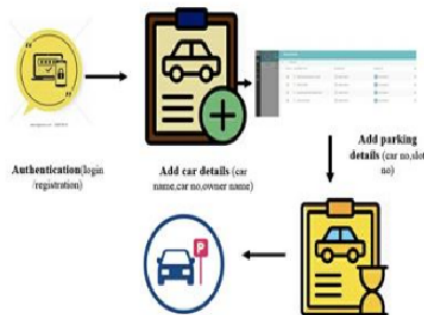


Fig 1: Architecture of the Proposed Model

It uses the method of vector quantization which has originated from signal processing. K-means clustering focusses to split 'n' observations into k clusters in which every observation belongs to clusters having the mean which is nearest to it.

B. K-Means Algorithm

In machine learning, K-Means Clustering is understood to be an approach of unsupervised learning in the field of Machine Learning. Here it is used to address clustering problems. This algorithm does the job of separating 'n' observations into k clusters, by ensuring that each observation belonging to the cluster with the closest mean, which acts or serves as the leader of the cluster. Here K describes the amount of pre-defined clusters. To go more in depth, it identifies the centroids present and does the allocation of all the data points present to the cluster near to it. Therefore, a wide range of cells emerge.

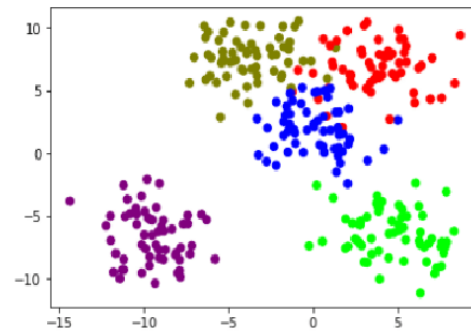


Fig 2: Representations of Clusters in K-Means Algorithm

C. Working of the Proposed Model

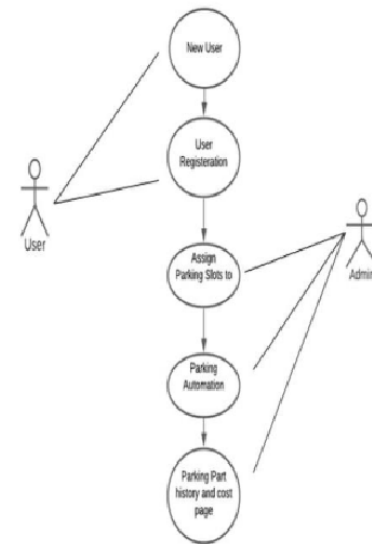


Fig 3: Workflow of the Proposed Model

Fig.3 shows the workflow of the proposed model. Since, machine learning model is built based on the car details given, it is required to create a form for Authentication in the

Visual Studio. Authentication here comprises of the login details of the user. Once we enter the login details, we are directed to the next page that is Car details. Car details comprises of our car no, name etc.

Once we are done with the details of the car, we are being moved to the next page which is the car parking details. In this form, we are asked to fill the data and time of the car to be parked, after which we are given the slot number. Using all these data given by the user, a Machine Learning model is being built using the K-Means Clustering Algorithm. K-means clustering algorithm is an Algorithm which is effectively used for predictions and is very helpful in building various Machine Learning Models. We have also used Google Colab for finding the accuracy of the K-Means Algorithm using Python.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section depicts the experimental results and discussion of the proposed work. The work is categorized into various modules as discussed below.

A. Master Login



Fig 4. Master Login

This master login module is developed for passengers, where a user logging into his/her own account will view this panel. The operations included in this module are,

- 1) Admin view Data – Here entire user data is been accessed by the person who controls the parking in the system.
- 2) User view Data after Anonymization – Here user cannot see entire fields of data like money earning, cost per hour, car availability etc., Anonymization means part of the data to be viewed by the user.

B. Company Owner Login

This module is only accessed by the owner or the proprietor of the parking space area. This is handled is particularly handled by him/her as it contains confidential details such as cost, initial cost etc. This is not dealt with any other person other than the owner himself. Hence this module is only accessed by him/her.



Fig 5. Car Parking Layout

C. Admin Login

Admin module is specifically developed for the people who have the rights to store the details of the car into the database. Here they will be storing the details of the car in order to get the car a slot in the parking slot section.



Fig 6. Administrator Login

D. New User Registration

When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. A person has to fill the details and submit. The details consist of firstname, lastname, password, Security Question, Security Answer, Mobile Number and city. This is done in order to get the vehicle registered.



Fig. 7 New User Registration

When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. When the person is new to the registration, he/she has to click New User. Once it gets clicked, it will reach new user registration page. A person has to fill the details and submit. The details consist of firstname, lastname, password, Security Question, Security Answer, Mobile Number and city. This is done in order to get the vehicle registered.

E. Adding Vehicle Details

Once the user logged into the system, they can add the vehicles and its details. Such as,

- 1) Owner Name
- 2) Vehicle Number
- 3) Owner Driving license number
- 4) Vehicle model
- 5) Date and Time of Parking
- 6) Number of hours for Parking

7) Any special parking credentials like offers etc.,

Fig. 8 Vehicle Details

These details are added for uploading in the database. These details will be stored and will be used for allocating the spaces with respect to the order it has been stored. These details are updated and stored by the admin, which is necessary for the successful allocation of the car parking space.

F. Car Parking Slot

Based on the date and timing of the parking, an automatic allocation is done with the K-Means algorithm. In this module, a clear view of the cars allocated can be visualized with the text bounded on each location. A complete visualization of the car parking slots will be displayed on screen. This page provides a complete picturization in the grid format on all the parking allocation slots. A list of unallocated locations will be displayed. In addition, a clear picturization of the user views and their display data can be visualized in this module. This module provides a complete control of working in the system.

Fig. 9 Car Parking Slot Module

G. Car Allocation and Cost Estimation

This module can be accessed from the user. It provides a clear picture on the past history of the vehicle. Any pending amounts related to the car and any extra charges which needs to be incorporated should be manipulated in this module. Any valet parking cum offers associated with the customer needs to be manipulated will be added in this module. As a result, the cost estimations would be displayed here.

The result shown in fig. 11 allocates the space in an orderly manner. The vehicles are also being updated in order to occupy the space in a proper manner. The K-Means Algorithm is implemented here and this helps in allocating the space in a systematic way. The details of the car, the car owner etc are being displayed on the side. There is also costs

being assigned to each car to which users have to pay. This is entirely been done in Visual studio. Every icon is being designed using Visual Studio and the language used here is c#. also, we are using SQL server as on our backend for the database connectivity.

Fig 10. Car Allocation and Cost Estimation

Fig 11: Allocation of Parking Spaces in a given order

V. PERFORMANCE ANALYSIS AND COMPARISON

This section shows the performance analysis of the proposed system with the existing approaches. The proposed approach identifies the free slot for parking more efficiently. Here each parking slot is scanned and as a result, nodes are generated in order to reach the destination. This algorithm is known for its best way of finding the shortest path. But since there are a lot of calculations involved in this process, it is quite uncertain to find all the shortest paths if given an empty space.

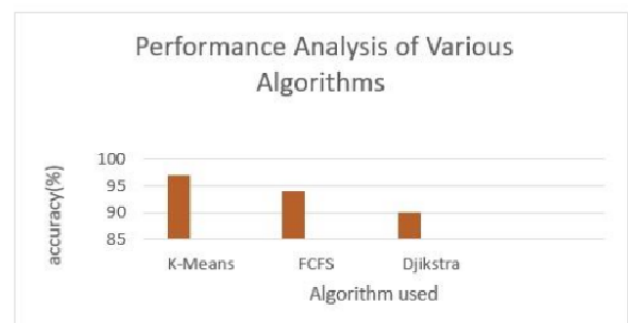


Fig 12. Performance Analysis

The implementation of the above system would help in predicting the space accurately and allocates space generously using the machine learning algorithms. In this paper, K-Means Algorithm is used in predicting spaces and also is useful in allocating them in a hassle-free manner. It can save time by not causing traffic and other kind of congestion. Also, by not wasting time as drivers find it hard to find the space when it is mostly occupied. In future, it can be achieved using various other algorithms which will be more precise and advanced in prediction by using the data given to it as that is the ultimate purpose of Machine Learning.

In this Comparison shown in fig.12, the representation of various algorithms is found in terms of its accuracy. Here, the accuracy is calculated by the sum of squared errors. K-Means is leading because it has 97% Accuracy in terms of performing and giving accurate results or Output. Also it is fast in processing data. K-Means is one of the most sought after algorithms in Machine Learning as it helps in grouping data in clusters and providing the desired outputs. It is used in prediction, allocation etc. It is an iterative algorithm partitions the dataset into K predefined distinct non overlapping subgroups which is otherwise called Clusters. And most importantly it is an unsupervised learning algorithm. Hence, it is efficient in this system being proposed here.

The next algorithm is FCFS which has an accuracy of 94% in terms of its performance. Here, the algorithm works on the process of First come First Serve basis. The purpose of using the algorithm is for allocating the car in a queue manner. To be more precise, this algorithm is used for Queue allocation of Cars. Since the working of Queues is based on First Come First Serve basis. As an algorithm it helps in allocating the cars as it has to find the spaces and allocates the car one by one. As a result it is good in performing and yielding accurate results but it lags when compared to K-Means Algorithm. To make an analysis, it is one of the second most performing algorithms after K-Means. It just makes a 3% difference from K-Means Algorithm. And the last one is Dijkstra Algorithm which has an accuracy of 90%. Here Dijkstra Algorithm helps in finding the shortest path. In a scenario like parking allocation this algorithm finds the shortest distance of the parking space and allocates the space in a proper manner. Under this it takes into consideration many factors like Geographic Information System (GIS), Global positioning System (GPS).

VI. CONCLUSION

In this paper, an efficient scheme for parking slot allocation is provided. The analysis shows that the proposed K-Means clustering method provides better accuracy compared to existing methods. In future, the scheme can be extended by adding additional attributes so, that the accuracy can be further improved.

REFERENCES

- [1] Nyambal, J., & Klein, R. (2017, November). Automated parking space detection using convolutional neural networks. In 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech) (pp. 1-6). IEEE.
- [2] Gupta, R. K., & Rani, G. (2020, March). Machine Learning and IoT based Real Time Parking System: Challenges and Implementation.

In Proceedings of the International Conference on Innovative Computing & Communications (ICICC).

- [3] Siddiqui, S. Y., Khan, M. A., Abbas, S., & Khan, F. (2020). Smart occupancy detection for road traffic parking using deep extreme learning machine. *Journal of King Saud University-Computer and Information Sciences*.
- [4] Ali, G., Ali, T., Irfan, M., Draz, U., Sohail, M., Glowacz, A., ... & Martis, C. (2020). IoT based smart parking system using deep long short memory network. *Electronics*, 9(10), 1696.
- [5] Srivastava N., Hinton G., Krizhevsky A., Sutskever I. and Salakhutdinov R., Dropout: A Simple way to prevent Neural Networks from overfitting, *Journal of Machine Learning Research*, Springer 2017.
- [6] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: path planning and control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp.396-410, 2015.
- [7] F.Bock, S.Di Martino, and M. Sester, "Data-driven approaches for smartparking," in *Machine Learning and Knowledge Discovery in Databases* Springer International Publishing, 2017, pp. 358-362
- [8] Shree, B. (2017). Design and Implementation of Automated Car Parking System using RFID. *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 6(2), ISSN: 2319 5940
- [9] Zips, M. Böck, and A. Kugi, "Optimisation based path planning *Robotics and Autonomous Systems*, vol. 79, pp. 1-11, 2015.
- [10] Z. Mahmood, "Towards a fully automated car parking system," *IET Intelligent Transport Systems*, vol. 13, pp. 293-302(9), February 2019.
- [11] Bevis Jinila, K.Komathy (2015). Rough set based fuzzy scheme for clustering and cluster head selection in VANET. *Elektronika Ir Elektrotechnika*, 21(1), 54-59.
- [12] Jinila, B. (2015). Cluster oriented ID based multi-signature scheme for traffic congestion warning in Vehicular Ad hoc Networks. In *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2* (pp. 337-345). Springer, Cham.
- [13] Bevis Jinila, K.Komathy (2015). Rough set based fuzzy scheme for clustering and cluster head selection in VANET. *Elektronika Ir Elektrotechnika*, 21(1), 54-59.
- [14] Jinila, Y. B., Sheeba, G. M., & Shyry, S. P. (2021). PPSA: Privacy preserved and secured architecture for internet of vehicles. *Wireless Personal Communications*, 118(4), 3271-3288.

Smrithii_Nithya_19-03-2022_IAIM 2022.doc

ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

5%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

downloads.hindawi.com

Internet Source

1%

2

"KSE 2019 Conference Proceedings", 2019
11th International Conference on Knowledge
and Systems Engineering (KSE), 2019

Publication

1%

3

ymerdigital.com

Internet Source

1%

4

Submitted to Wawasan Open University

Student Paper

1%

5

Submitted to Bowie State University

Student Paper

1%

6

Submitted to University of Auckland

Student Paper

<1%

7

repository.tudelft.nl

Internet Source

<1%

8

Hong Xu, Yuan He, Xiang Sun, Jia He,
Qiongmei Xu. "Prediction of thermal energy
inside smart homes using IoT and classifier

<1%

ensemble techniques", Computer
Communications, 2020

Publication

9

scholar.google.com

Internet Source

<1%

10

Cushil Kumar Singh, Yi Deng, Long Huang, Dark

1%

10 Sushil Kumar Singh, Yi Pan, Jong Hyuk Park. "Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City Environment", Sustainable Cities and Society, 2021
Publication

11 ijariie.com
Internet Source

12 www.0517tv.com
Internet Source

13 Walter Balzano, Marco Lapegna, Silvia Stranieri, Fabio Vitale. "Competitive-blockchain-based parking system with fairness constraints", Soft Computing, 2022
Publication

14 link.springer.com
Internet Source

15 spig.upi.edu
Internet Source

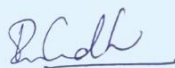
16 www.amrita.edu
Internet Source

17 Sunil Kumar. "Security Enhancement in Mobile Ad-Hoc Network Using Novel Data Integrity Based Hash Protection Process", Wireless Personal Communications, 2021

Exclude quotes On
Exclude bibliography On

Exclude matches Off

Presentation Certificate:

	SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY) Accredited with Grade 'A' By NAAC 12B Status by UGC Approved By AICTE www.sathyabama.ac.in	  IEEE Advancing Technology for Humanity
School of Computing Department of Information Technology Certificate of Presentation		
This is to certify that SMRITHII. R of Sathyabama Institute of Science and Technology has presented a paper entitled		
An Efficient Parking Allocation Scheme Based on K-Means Clustering Method in the AICTE sponsored International Conference on Artificial Intelligence and Machine Learning (IAIM-2022), held from 27th January 2022 - 29th January 2022.		
 Dr. R. SUBHASHINI HOD [Information Technology]	 Dr. T. SASIKALA Dean [School of Computing]	 Dr. T. SASIPRABA Vice Chancellor