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

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Winter Semester 2021-22

Grook - A multiservice sports platform optimized using Machine Learning

Done by:
Neel Rakesh Choksi 19BCE0990
Khushee Jain 19BCE0914
Panshul Jindal 19BCE2227

Course Code: CSE1901
Course Name: Technical Answers for Real World Problems

Under the Guidance of:

Prof. Amutha Prabhakaran M

Abstract

Participation in sports is extremely important, and should be encouraged. The main benefits of sports are improved health and fitness, learning how to collaborate in teams, develop social networks and also work as a stress buster. However, it is not always easy for users to access these facilities. Also the ground owners and facility providers do not make much profit because of lack of marketing and publicity. This sector does not have a centralized system to solve such problems.

Our goal is to develop an android mobile application that will make it easier for users to find sports facilities in their vicinity, book convenient time slots and also form teams with other users with similar sports interests. Based on the user's preferences, the suggestions will be optimized and personalized using different ML algorithms and models in the areas of Regression , Association rule mining , Sentiment classification to improve user experiences and Maximize owner profits.

Introduction

Despite being familiar with all the pros of sports. In our country there is a large gap between the facility providers and the people having interests.We wish to bridge this gap and design a centralized system to make it easy for the passionate people as well as for the facility providers. We will focus on building an android application integrated with ML models so that the facility can be accessed by anybody anywhere and ML models will increase the efficiency, user experience, productivity and maximize facility providers profit based on their preferences. In the application user can easily browse grounds based on their locality and the app will display the ground based on which sports is liked by the user, the ground preferred by the most of the users, etc.The app will allow you to see the list of players already booked the ground in the same slot so that players can form a team and enjoy the experience. To achieve our aim we will create a database that contains all details of the registered sports grounds. These details will then be mapped to the user’s needs so as to help him find a sports facility in his vicinity at a convenient time slot. Also the users can view their complete booking history and book the grounds directly from there. For the facility owners the ML model will predict the amount of people who will book various sports and grounds on the particular time slot so that they can manage the facilities accordingly. Each user will have a unique login to ensure authenticity.

Problem Statement

Generally citizens are unaware about the sports facilities provided by a sports club or by the nearby playgrounds. People are oblivious to which ground provides good services and which has poor management. It becomes a challenge for the players to distinguish different facilities based on services & qualities and compare prices. Further, predicting the time to play along with more players is a tough task as it's not easy to find people having the same sports interest. Ground owners are not able to make a worthy profit and the pandemic has led to more losses and weak management as they cannot anticipate the number of players on a particular day for a particular sport.There is a need for a system to solve this issue so that user can choose among the facility and get the suggestion based on their interest and also the owners can manage the slots and maximize their profits if they can priorly predict the number of bookings they will get on a particular day within a certain time slot.

Objective

Our application will make it easier for users to find sports facilities in their vicinity, book convenient time slots and also form teams with other users with similar sports interests. To achieve our aim we will create a database that contains all details of the registered sports grounds. These details will then be mapped to the user's needs so as to help him find a sports facility in his vicinity at a convenient time slot.
Analytics to be done for customers showing the frequency of visiting a ground, choosing a particular sport, time slot , cost spent on a ground, time slot, sport. Analytics to be done for ground owners showing the number of people visiting, sports chosen, time slot chosen on a day, revenue generated from a sport, time slot , on a day and overall revenue.

1. Predicting ground, sport and time slot of a user based on previous bookings.(customer & owner) using **association rule learning.** [Apriori, FP growth, ECLAT]
   [Linear, Multiple, Ridge, Neural Net, Lasso ,Decision Tree, Random forest, KNN, SVM]
2. Predicting number of customers for a particular ground & slot (Owner) using **regression**
3. Revenue prediction based on day, sports & time slots (owner) using **regression**
4. Rate grounds based on customer reviews(owner) **classification** [Naive Bayes, SVM , LSTM, GRU]

Literature Review

[1] Crypto-Currency price prediction using Decision Tree and Regression techniques

The aim is to consider all influencing parameters of Bitcoin price to forecast the Bitcoin price precisely and use different machine learning algorithms for prediction and compare their results. Firstly it identifies the trends of day by day changes in the prices.Dataset consists of open, low, high and close values of bitcoin price. Data published in Quandl.com is exploited with name mentioned as “BITSTAMPUSD”. The data collected with following features and stored as data.csv “Time_stamp, Open, High, Low, Close, Volume_btc, Volume_currency, Weighted_price then the dataset is pre-processed. After this dataset is splitted as train and test set. Train the model after applying decision trees and regression and at last test values are given. 80% of data is considered as training input for our machine learning algorithm model to train the model. The remaining 20% of data is considered as a test for result prediction. The paper exploited Lasso and Regression to predict the price trend for 20% test input and the predicted values were plotted and compared for accuracy. The result shows that price prediction is efficient using linear regression algorithm and achieves 97.5% accuracy whereas decision tree achieves 95.8% accuracy. This outperforms the accuracy of existing works. The experimental study results show that linear regression outperforms the other by high accuracy on price prediction.

[2] Predicting Movie Box Office Success using Multiple Regression and SVM

The paper identifies the factors that are important for a movie’s huge profit and predicts the success of a movie which saves movie studios hundreds of millions of dollars a year. Efficiency is analyzed using multiple linear regression and Support Vector Machine by influence of variables like Wikipedia page views, time of release, critic ratings and trailer view. The dataset was populated from BoxOfficeMojo and Wikipedia. Trailer views were taken from YouTube. To paste the content in google sheets from the wikipedia page a script was enabled and then it was exported to Excel.From the resultant dataset,duplicate entrie and the movies having incomplete information or junk values had to be removed. Dataset is classified in 3 categories: x Low Budget Films (Budget < $50 million) x Medium Budget Films (Budget between $50 million and $150 million) x Big Budget Films (Budget >$150 million).The paper used different sets of publicly available relevant data and used multiple regression to come up with a predictor for movie success, with the resultant R-value greater than 0.88. SVM was used to train multiple variables and the result was an accurate classification rate of 56.52%, which is a higher accuracy than previous attempts at SVM using only a single variable.

[3] Sentiment Classification

Classification and Clustering is used to determine if the email is spam or not but to determine the precision of the various algorithms evaluation measures play a vital role. Here the evaluation measures are True Positive, True Negative, False Positive, False Negative. True positive metric is more when the model detects a spam email as spam .False positive metric is more when the model detects a non spam email as spam. True negative metric is more when the model detects a non spam email as non spam . False Negative metric is more when the model detects a spam email as a non spam.

The MIME(Multipurpose Internet Mail Extensions) parser was used to collect file name, email body , from , subject and the sending date for every email. The frequency of each word used in the corpus is calculated. They selected a threshold value for the frequency. Stemming is a process of reducing the word to its root form . Stemming was applied on the corpus. The common dictionary words such as a, the, in , I which occur very frequently in the text corpus are called stop words and they were also removed.Then the most frequently occurring words were used as document features.A matrix was formed with words as rows and frequency as columns. A random document was then selected and its similarity with every other email was calculated to determine its cluster.

The analysis of a text corpus of 19,620 emails is accumulated.The general statistics about the emails is also collected using google reports.These emails are further parsed using an external tool. The emails used were personal emails containing large amounts of contents. The emails can be classified using several natural language processing techniques and text parsing used for pre processing.The emails have to be classified for various reasons including spam, subject or folder classification and also for community detection.A large data set was integrated here and pre processed. Classification algorithms conducted gave a high percentage of True Positive which means that the email was spam and was also predicted as spam by the classifier.NGram based clustering gave the best results.

The main challenge was to handle a large data set. The number of unique terms available as an input for the text classifiers was huge.A future solution can be that a more efficient model can be created which is intelligent in nature and can classify the emails more effectively based on the previous experiences in real time.

[4] Sentiment Classification

The corpus is preprocessed and is done using various techniques namely Lexical Analysis also called tokenization . The given string of email body and subject is tokenized into words.Next the stop words are removed . Stop words are the words that occur too frequently and are non informative in nature.Example a, then , I, an ,it.Stemming is then performed on the data set

.Stemming helps reduce the data set words to its root form.Finally the words are represented in matrix form for processing it in the machine learning algorithm.The analysis of the data set can be done using feature extraction, feature selection or email header analysis. Feature extraction includes the words occurring in the email as a feature of that email.These features should now be selected according to relevance .It is done using classifiers.To determine the email as spam or not, the headers of the email are used. Headers of email determine the recipient of a message .The route of the email in the mail server can be used to determine if the email is spam or not.
Knowledge based filters can be used to determine the email as spam or not. These filters are based on coded rules or heuristics which compare the email with the occurrence of words such as lottery.It is difficult to maintain a set of rules that are effective in determining the email data .Since new spam techniques and issues are rising by the day.Another method used is by using the IP addresses of the server from which the mail was received.A blacklist is maintained in the DNS(Domain Name Server) System and every time a mail request arrives, a lookup in the blacklist table is carried out.Similarly whitelisting can also be carried out.Spam filtering is a binary classification task and Naive Bayes approach is suggested.It ignores the possible dependencies or correlations between the multivariable problem to a uni variable problem. Support Vector Machines can be used . SVM are a result obtained by mapping the feature set of vectors (training data) with a linear or a non-linear space through a kernel function. Furthermore clustering techniques can be adopted , text clustering using a vector space model can be adopted. Finally another method suggested is called ensemble classifiers which enhance the results by using a model on partitions of the training data and evaluating the outcomes.
The data sets suggested are the SpamAssassin dataset, Enron-Spam, LingSpam , GenSpam and the Spam Base.Content based spam filtering has shown the best results among the various methods of spam detection.Among the methods such as Heuristic filters, blacklisting, whitelisting, greylisting, challenge response systems, collaborative spam filtering , honey pots, signature schemes.
There cannot be a single solution for the detection of spam emails since the people spreading spam emails are finding out the loopholes in the classifiers.More research can be conducted on the clustering techniques and ensemble classifiers since they improve the performance of the model.

[5] NB

Rathod, S. B., & Pattewar, T. M. (2015). Content based spam detection in email using Bayesian classifier.The Internet provides Emails as means of data communication. Email messaging is an essential contribution. Hacking attacks, phishing attacks and malicious attack are frequently undergo email services to attempt fraud and deception motivation. They use emails to obtain personal credentials of user for financial gain. The set of labels are transformed into a format which the machine learning algorithms can compute classification is done based on the feature data set produced by the input data which helps in determining the classifier function.In content

based spam filtering, every message is represented as a binary vector in which the nth term in the vector shows whether it occurs in the document or not. Based on these performance parameters the probability of classifying the message as spam is calculated. large and real public data Enron datasets are used and a corpora was composed which has legitimate messages.This method shows the best results among the various methods of spam detection Among the methods such as Heuristic filters, blacklisting, whitelisting, collaborative spam filtering systems, honey pots,etc like random forests. Research can be conducted on the spamming techniques and ensemble classifiers since they improve the performance. Some spammers also ingest a lot of unwanted data in the email messages, a technique to overcome that should be found and a flexible way of comparing the factors.

[6] SVM

The Support Vector Machine is a type of supervised machine learning technique and is a recently developed framework, based on statistical learning theory. It has vast applications like it can be used in time series prediction , facial recognition, medical diagnosis. The problem of supervised machine learning is formulated by considering training data sampled according to a probability distribution and it also contains a function that calculates the error done by the model while training it.We need to find a function that minimizes the expectation of the error on new data(test data).The Support Vector machines find an optimal plane . The plane is said to be optimal since it is equidistant from all its nearest points known as support vectors. A plane is constructed in the case of a non linear data set. A line is constructed as a boundary when the data set is linear. The data points are classified based on a function f(x) . f(x) = w.x +b where w is the normal vector from the boundary , in the case of a plane it is the area vector of the plane. B is the offset . The plane induced by the model is formed using a Kernel function K(xi,x) which replaces the dot product between the vectors. The time series analysis of data can be done using SVM . The error function can be calculated automatically in this approach. The future values of the data set are predicted using SVM.An idea to split the training data into parts is suggested.This way many SVMs are trained rather than one global learning machine. This increased the performance of the model.SVM made for face classification had a kernel function designed by maximizing within the class variance. The third dimension is used to calculate the boundary parameters which are graphs of quadratic polynomials. Medical diagnosis was done for Tuberculosis from photomicrographs of Sputum smears. SVM is leaning towards statistical learning theories. The theory characterizes performance based on the ability to predict future data. The SVM is calculated using the Quadratic optimization techniques which involve the third dimension to classify points in the two dimensions.Training many local SVMs instead of training a global SVM is more likely to give the answer.Some methods for biasing the SVM towards a particular cluster were suggested.Standard SVM techniques can be modified for practical usage.The choice of the methodology to determine the kernel that determines the boundary is the area where further research can be conducted.If a group of SVMs are observed together then significant observations can be recorded.

[7] LSTM , GRU , Hybrid [Base Paper for sentiment classification]

Feedback on situations, events, products and services has risen due to the various social media platforms available . Sentiment classification is important to consider feedback in order to evaluate the product or service and possibly make improvements to it. Deep learning models including LSTM , GRU, BiLSTM , CNN are used in the paper to perform sentiment classification . The text representation methods used to input feedback into the models involve word embeddings which are helpful to find out the relation between words as words are represented as vectors and distance between them helps identify similarity .Various types of word embeddings were combined with different types of sentiment classification model , the combination providing the best accuracy was used to create a hybrid model . The authors propose a hybrid model combining word embeddings and sentiment classification models. Character level embedding was used to input text into the CNN model and FastText embedding was used to input text into BiLSTM model . The results of these two branches were combined into the final layer deciding the classification .

A total of 17289 Twitter tweets related to GSM communication service providers between 2011 and 2017 were aggregated. The sentiment classes were positive, negative, neutral . The tweets were divided into 13832 for training the model and 3457 for testing the model . Preprocessing of the feedback tweets is done by converting to lowercase, replacing links to url , replacing @usernames to usernames, removing whitespaces, characters (‘/’,’(’,’[’,’&’) , removing the punctuation , removing unrecognized characters, removing all digits.

Word representations play an important role in performance of sentiment classification models. The models perform better when the text is provided in a word embedding representation. The CNN(Convolutional Neural Network) can perform feature extraction in local regions, while the LSTM(Long Short Term Memory) network can perform better with datasets having long term dependencies in the sentences . Word embedding methods include Word2Vec , FastText . In this paper , the authors have represented the text using FastWord and character-level embeddings . The text is fed into two branches to classify the sentiment. In one branch , the text is represented as character level embeddings and fed into the CNN neural network for feature extraction . In the other branch , the text is represented in word embeddings using the Word2Vec and FastText model and fed into (LSTM/GRU/BiLISTM) for feature extraction .The features extracted from both the branches are fed into another layer where they are concatenated . Finally Sentiment of the feedback is found in the last layer . Attention mechanisms can be incorporated in the hybrid model to improve its performance.Stemming , lemmatization , stop word removal can also be performed to preprocess the tweets.

[8] Recommendation System using Apriori Algorithm

Recommendation System using Apriori Algorithm, 2015

This paper presents a new recommendation technique using the Apriori algorithm. Its aim is to detect association rules. "In 70% of the cases, when a customer purchases bread, he also purchases butter or jam." The algorithm looks for things that are commonly purchased and then suggests them to the customer as a recommendation.
Groups of candidates (the candidate set contains all the frequent k-length item sets) are tested against the data. The algorithm terminates when no further successful extensions are found. A recommendation system is a new interactive technology that allows any organization to obtain more data from its transaction-oriented database of clients. This approach assists clients in locating things on the site that they wish to purchase. The most significant difficulty is the time spent holding a large number of candidate sets with frequent itemsets, low minimum support, or large itemsets..
Recommendation system holds a strong future in Ecommerce on the web. This recommendation system will be used at numerous sites to recommend various services in the future. The technique could be extended to web content mining, web structure mining, and other applications in the future. It is also possible to extend the work to extract data from image files.

[9] Hybrid based recommendation system based on Clustering and Association mining

A Novel Hybrid based Recommendation System based on Clustering and Association Mining, 2016

A recommendation system filtered the data using data analysis techniques, allowing the user to recommend the most appropriate goods. Recommendation systems use a certain sort of information filtering system technique to suggest information items (movies, TV shows/episodes, music, books, news, photos, web pages, scientific publications, and so on) that are likely to be of interest to the user.

Recommendation systems are classified into 3 approaches which are collaborative, content-based or knowledge-based methods to have a better recommendation.
A. Collaborative based Recommendation systems
The collaborative filtering Algorithm recommender system has become one of the most studied recommender system strategies. If individuals previously shared similar interests, they will have comparable tastes in the future.

B. Recommendation System Based on Content It deals with user profiles that were built at the start in content-based recommender systems. A profile contains information about the user and his or her preferences, which are determined by how the user ranks goods.
Hybrid Recommendation System (Hybrid Recommendation System) (C) Hybrid Recommendation is a hybrid strategy that combines both collaborative and content-based approaches. Different types of challenges can be easily overcome with Hybrid Recommendations, such as the Cold-Start problem.

Using clustering, you may improve the diversity, consistency, and reliability of suggestions, as well as the data sparsity of user-preference matrices and changes in user preferences over time. The future work is to improve the accuracy of the system by enhancing the clustering algorithm.

[10] Expert recommendation algorithm based on Pearson correlation coefficient , FP-Growth

An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth, 2018

Automated allocation of the project to the most appropriate expert examiners has become a hot topic for many researchers. The traditional expert recommendation system mainly focuses on analyzing the research direction of experts based on a large number of historical information sets, quantifying expert expertise and conducting expert classification, matching experts and projects based on project attributes and expert types.

(1) Preprocessing of historical project information data and historical project review expert data.
(2) Obtaining frequent item set of review experts by using FP-growth method on the preprocessed historical project review expert data
(3) Calculating the Pearson similarity between each project of historical project information data and the recently project to be reviewed
(4) According to the above Pearson similarity, selecting the most similar historical project with the recently project to be reviewed from historical project information data.
(5) Selecting the experts set of the above most similar historical project obtained in the step 4.
(6) Based on the review direction of each expert in the experts set and the type of project to be reviewed, a candidate expert set are produced by combining the experts obtained by the step of 5.
(7) According to frequent item set of review experts obtained by the step 2, computing the combination frequency and coincidence degree of each candidate expert set.
(8) Selecting the candidate expert set as the final recommended expert set which coincidence degree is the highest.

Effective recommendation of a drawing review expert combination with the highest degree of fitness makes the experts to improve the efficiency of collaborative review, and increases the use value of history project review expert combination datasets. The recommend algorithm proposed by this paper can be used for reference in some population recommendation, its improvement and optimization for applicable to different recommendation environments are future study direction and further improvement of the accuracy of the algorithm also is a direction for future research.

[11]Author age prediction from text using linear regression

Author Age Prediction from Text using Linear Regression, 2011

A person is a member of a multiplicity of communities, and thus the person's identity and language are influenced by many factors. In this paper they focus on the relationship between age and language use. Recently, machine learning methods have been applied to determine the age of persons based on the language that they utter.

The first contribution in this paper is an investigation of age prediction using a multi-corpus approach. We present results and analysis across three very different corpora.A second contribution is the investigation of age prediction with age modeled as a continuous variable rather than as a categorical variable. Most prior research on age prediction has framed this as a two-class or three-class classification problem

The Fisher corpus contains transcripts of telephone conversations. People were randomly assigned to pairs, and for every person, characteristics such as gender and age were recorded. They drew data from one of the most active online forums for persons with breast cancer. All posts and user profiles of the forum were crawled in January 2011. Only a small proportion of users had indicated their age in their profile. We manually annotated the age of approximately 200 additional users with less common ages by looking manually at their posts

Given an input vector $x \in R$ m, where x1, . . . , xm represent features (also called independent variables or predictors), we find a prediction $\hat{y} \in R$ for the age of a person $y \in R$ using a linear regression model: $\hat{y} = \beta 0 + x > \beta$ where $\beta 0$ and β are the parameters to estimate. Usually, the parameters are learned by minimizing the sum of squared errors.

We presented linear regression experiments to predict the age of a text's author. As evaluation metrics, we found correlation as well as mean absolute error to be complementary and useful measures. We obtained correlations up to 0.74 and mean absolute errors between 4.1 and 6.8 years. In three different corpora, we found both content features and stylistic features to be strong

indicators of a person's age. By learning jointly from all of the corpora, we were able to separate generally effective features from corpus-dependent ones.

[12]Heart disease prediction using hybrid machine learning model

Heart Disease Prediction using Hybrid machine Learning Model, 2021

Heart disease causes a significant mortality rate around the world, and it has become a health threat for many people. Early prediction of heart disease may save many lives by detecting cardiovascular diseases like heart attacks etc., is a critical challenge by the regular clinical data analysis. Machine learning can bring an effective solution for decision making and accurate predictions. The proposed study used the Cleveland heart disease dataset, and data mining techniques such as regression and classification are used. Machine learning techniques Random Forest and Decision Tree are applied.Three machine learning algorithms are used in implementation:

1. Random Forest: Random Forest regression aggregates multiple decisions to make a single decision. For training characteristics and then random sub characteristics for sampling nodes, random sampling is done.

2. Decision Tree: Decision tree is one of the learning models that is used in the problem of classification. We divide the dataset into two or more sets using this technique. In decision tree, internal nodes represent a test on the characteristics, the branch portrays the outcome, and leaves are the decisions generated after subsequent processing.

3. Hybrid model (Hybrid of random forest and decision tree): We develop a hybrid model using a decision tree and random forest algorithm. The combined model works based on probabilities of random forest. The probabilities from the random forest are added to train data and fed to the decision tree algorithm.

The study aims to predict heart disease based on machine learning via an automated medical diagnosis method. A hybrid model is a novel technique, which uses the probabilities arrived from one machine learning model as input to the other machine learning model. This hybrid model gives us better-optimized results based on both machine learning algorithm, which is considered for the implementations.

In the future as Deep learning algorithms play a vital role in health care applications, applying deep learning procedures for heart disease prediction may give better outcomes.

[13] Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection

Performance of intrusion detection systems depends upon the accuracy to detect false alarms and increase their detection rates but there's so much of improvement required to increase accuracy. Thus this paper compares the approaches like SVM, random forest and extreme learning machine to find the most suitable technique.The key phases of the proposed model include the dataset, preprocessing, classification, and result evaluation. Each phase of the proposed system is important and adds valuable influence on its performance. The dataset is collected as sanitized dataset, simulated dataset,testbed dataset, and standard dataset.. However, complications occur in the application of the first three methodologies. A real traffic method is expensive, whereas the sanitized method is unsafe. To overcome these difficulties, the NSL–KDD dataset is used to validate the proposed system. Non-numeric or symbolic features are eliminated during pre-processing that generates overhead thus more training time and the architecture of classifiers becomes complex which wastes memory and computing resources. SVM, RF, and ELM are applied based on their proven ability in classification problems. Details of each classification approach are provided. ELM outperforms other approaches in accuracy, precision on full data samples that comprise 65,535 records of activities containing normal and intrusive activities. Thus ELM is a suitable technique for intrusion detection systems that are designed to analyze a huge amount of data like network data with huge traffic. In future, ELM can also be explored further to investigate its performance in feature selection and feature transformation techniques.

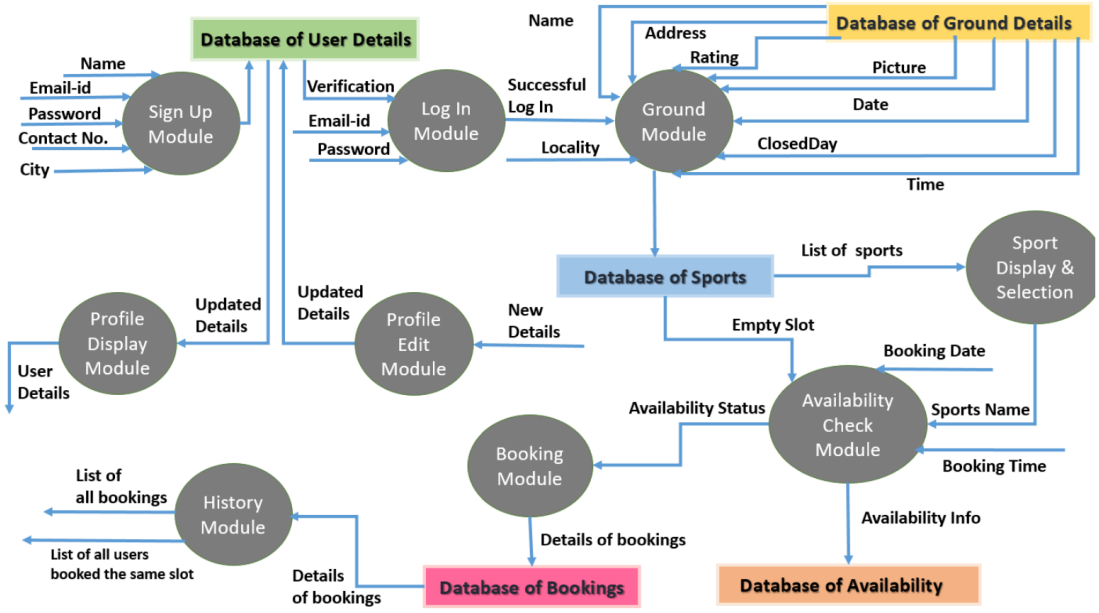
[14] High Precision Error Prediction Algorithm Based on Ridge Regression Predictor for Reversible Data Hiding

This paper presents a high precision error prediction algorithm a ridge regression based for reversible data hiding. It is a penalized least-square algorithm, which solves the overfitting problem of the least-square method. The residual sum of squares between predicted and target pixels is minimised using reversible data hiding based on ridge regression predictor, subject to a restriction defined in terms of the L2-norm. The ridge regression-based predictor may obtain more modest prediction errors than a least-squares-based predictor, demonstrating that the proposed method is more accurate. Additionally, the target pixels' eight neighbours, as well as their two different combinations, are chosen as training and support sets, respectively. This selection strategy enhances forecast accuracy even more. The suggested strategy outperforms state-of-the-art adaptive reversible data concealing in terms of prediction accuracy and embedding performance, according to experimental results. In future an optimization algorithm can be designed to obtain more accurate ridge coefficients and improve the accuracy of ridge regression predictor.

[15] Facial expression recognition using general regression neural network

Facial expression identification is described in this research employing an Efficient Local Binary Pattern (LBP) for feature extraction and an artificial neural network (ANN) for classification. The Efficient LBP algorithm is used to generate facial feature vectors from image dataset blocks of four different sizes (256 x 256, 128 x 128, 64 x 64, and 32 x 32). ANN is used to perform multiclass categorization of the image dataset into the six fundamental universal emotions. In this experiment, we used a General Regression Neural Network (GRNN) for classification. The network is trained faster with GRNN, which does not require an iterative training approach.It can be observed from the experimental findings, the network's free parameter, spread constant, is tailored to lower the mean square error and therefore enhance recognition efficiency. The proposed approach is evaluated utilising widely used standard databases, including the Japanese Female Facial Expression Database, the Taiwanese Facial Expression Database, the Cohn-Kanade Expression Database, and the Indian Student Face Database. The suggested approach enhances the identification rate by using 64x64 window sizes as the optimal window size.

Modules



Sign Up / Login

REQ-1: Mandatory enter the required fields - personal details i.e. Name , Address, Contact number, Email -id, Password

REQ-2: Password should follow the syntax of minimum one Upper Case letter, minimum of 8 characters and at least a special character.

REQ-3: The user needs to re enter the data if there is an error or an invalid input

User Profile , Reset Password

REQ 1: Updating details with valid information. System will ask to re enter the details if there is any error or invalid input

REQ-2:The Link for resetting password should be sent to the email-id of the user.

REQ-3: The new password should be framed such that it fulfills all the constraints mentioned.

REQ-4: System should update the database with the new details.

Checking availability

REQ-1: Selecting a suitable locality

REQ-2: Selecting a date and time

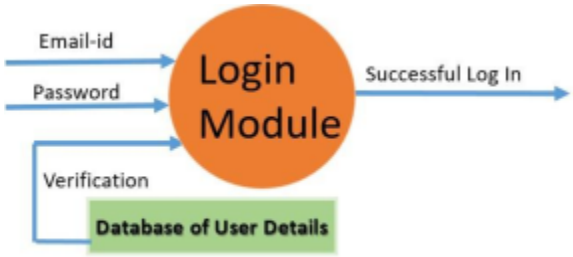
Booking and payment history

REQ-1: User should check the details before confirming the booking.

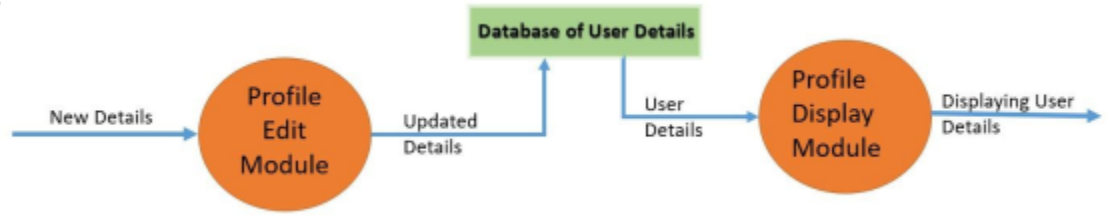
REQ-2:User should enter proper banking details to ensure the payment process is a success

Sign Up

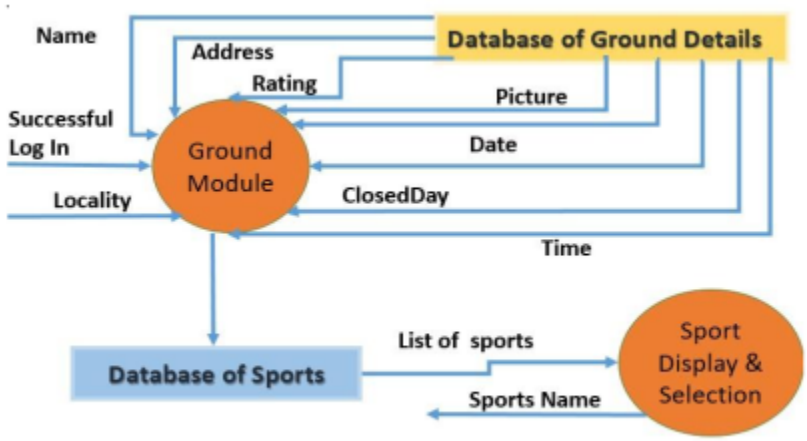
Login

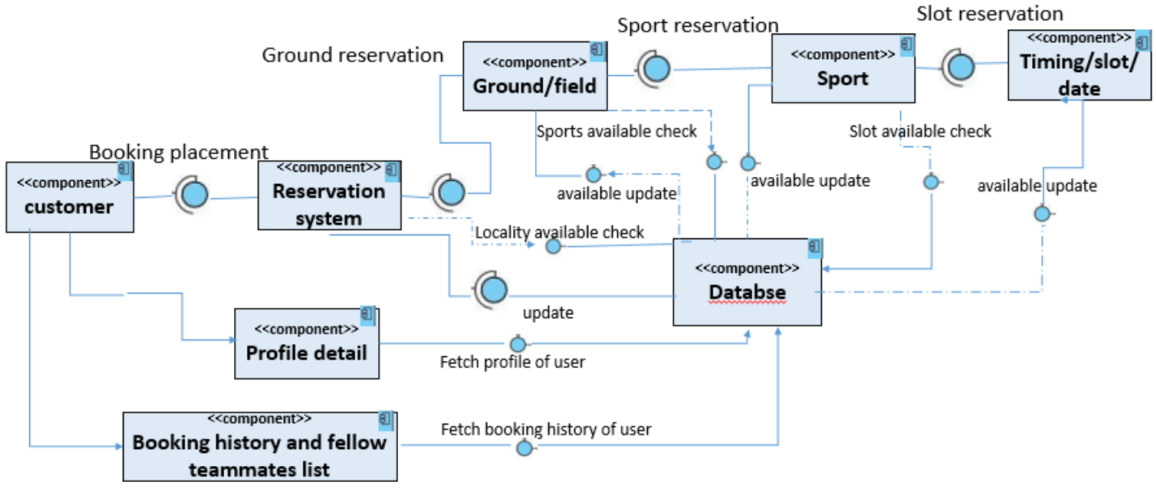


Profile



Ground

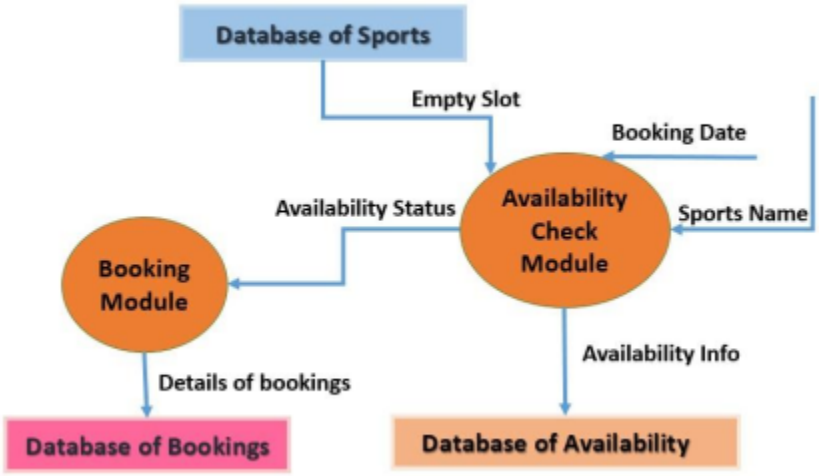




Based on the Database of Ground details the following analytics will be done by using bar charts and line charts. Bar charts would be formed for the number of people visiting a ground daily, monthly , yearly, number of people choosing a timeslot from available time slots ,revenue made in that timeslot , number of people choosing a sport from the available sports , revenue made from that sport. Line charts would be formed for number of people choosing a particular sport , revenue on that sport on daily,monthly, yearly basis , number of people choosing a particular timeslot , revenue on that timeslot on daily,monthly, yearly basis. Measures of central tendency to be applied for total revenue generated daily, monthly , yearly.

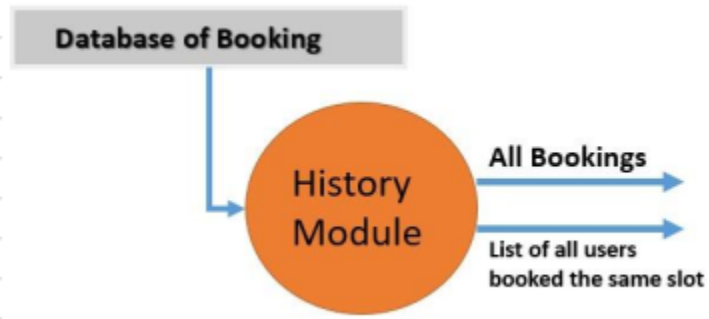
A prediction can be made for determining the most occurring combination of a particular sport and timeslot using association rule mining .Predictions can be done using regression. The daily,monthly,yearly number of people visiting the ground can be predicted . The number of people choosing a particular sport , timeslot can be predicted . The daily , monthly , yearly total revenue can be predicted. The revenue generated from a particular sport, timeslot can be predicted.

Booking



A prediction can be made for the most frequently occurring combination of selected ground, sport and timeslot based on previous bookings of all customers.It can be achieved by using Apriori Algorithm , ECLAT , FP growth algorithm for association rule mining.

History



Based on the booking history of the user, analytics will be made including bar charts and measures of central tendency. Bar charts will be made for representing frequency of user visiting a particular ground vs grounds , frequency of a user choosing a sport vs sports , frequency of a user choosing a timeslot vs all time slots. Measures of central tendency will be evaluated for cost spent by user on a ground, on a timeslot , cost spent daily , monthly and yearly. Also the total cost spent daily, monthly,yearly.

The cost spendings of the user can be predicted using a regression model.

User reviews

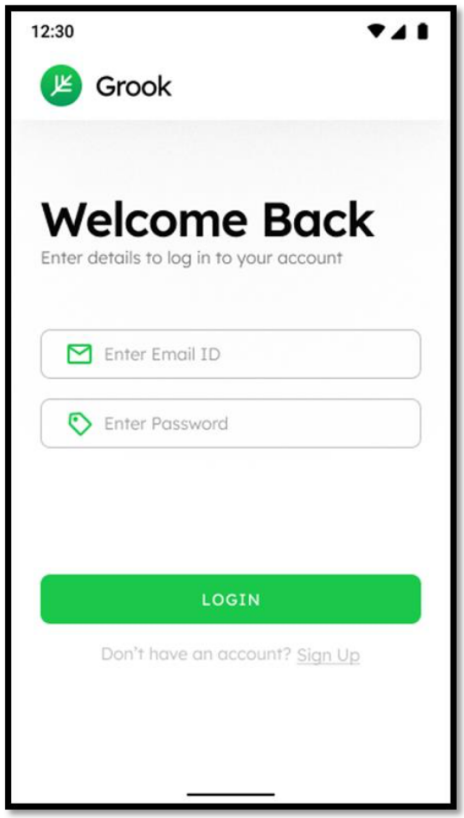
Users will be asked to give a written review for each ground. These reviews could be preprocessed, represented in the form of text embeddings and classify them into good , medium or bad feelings for the ground. Based on the classification , an average rating for the ground could be found.

The deep learning models to be used to achieve classification of reviews include Naive Bayes model , Support Vector Machine model , recurrent neural network models including LSTM(Long Short Term Memory) , GRU(Gated Recurrent Units).

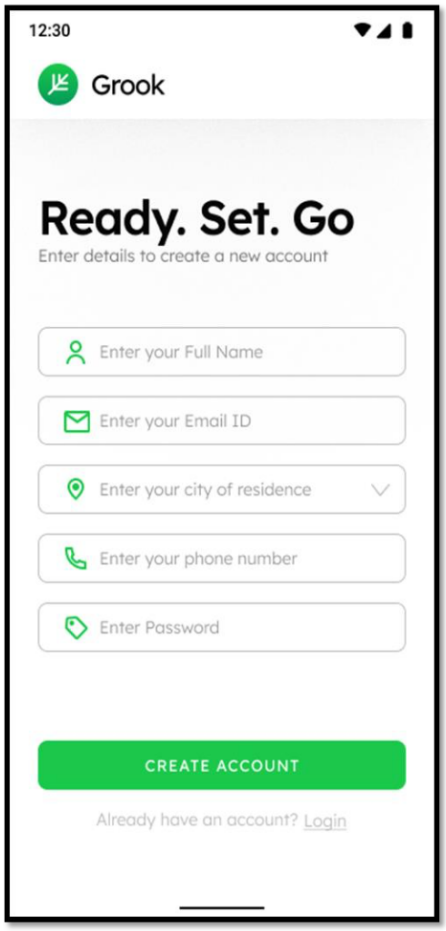
UI Design

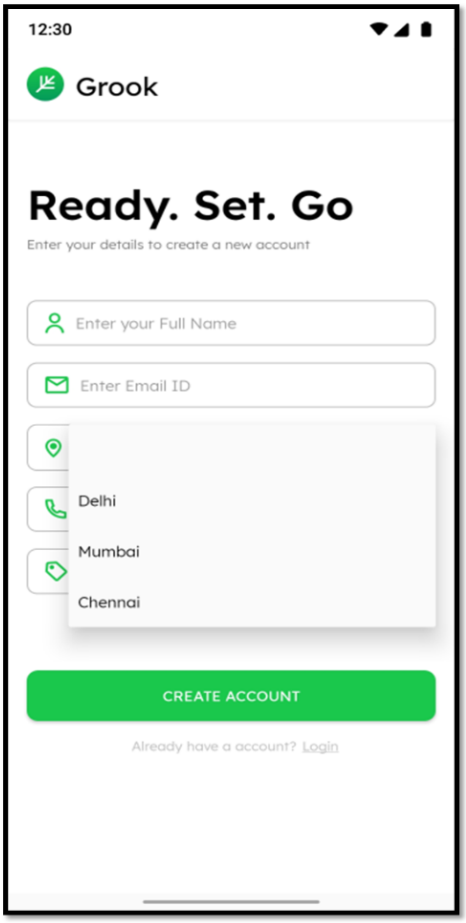
Landing Page

Login Page

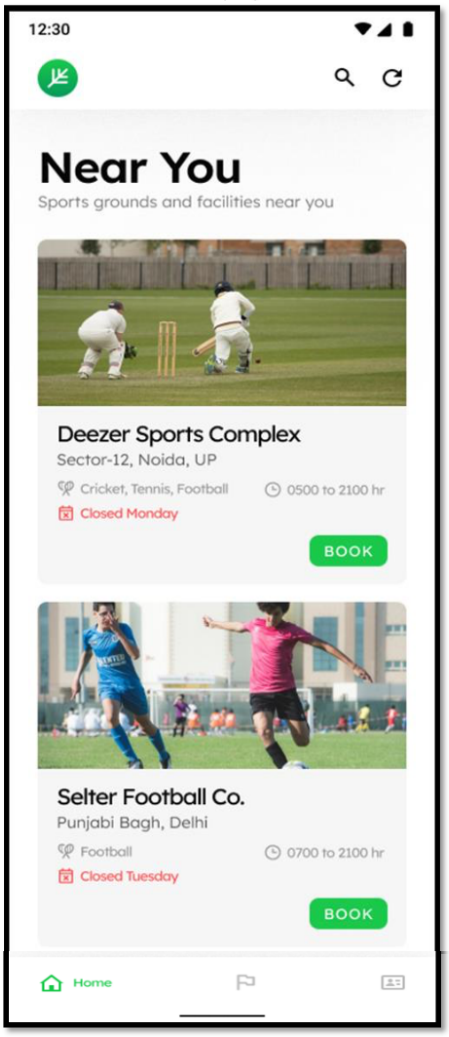


Sign up Page

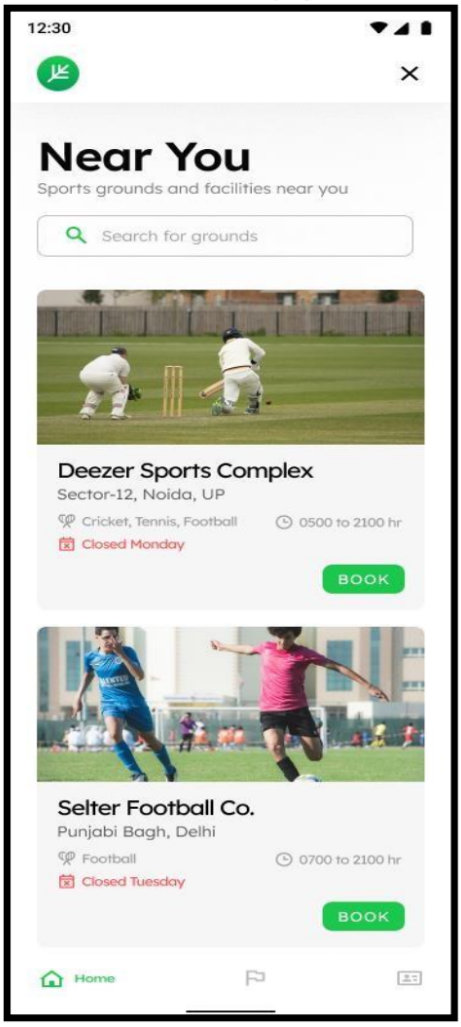

12:30
Grook
Ready. Set. Go
Enter details to create a new account
Enter your Full Name
Enter your Email ID
Enter your city of residence
Enter your phone number
Enter Password
CREATE ACCOUNT
Already have an account? Login


12:30
Grook
Ready. Set. Go
Enter your details to create a new account
Enter your Full Name
Enter Email ID
Delhi
Mumbai
Chennai
CREATE ACCOUNT
Already have a account? Login

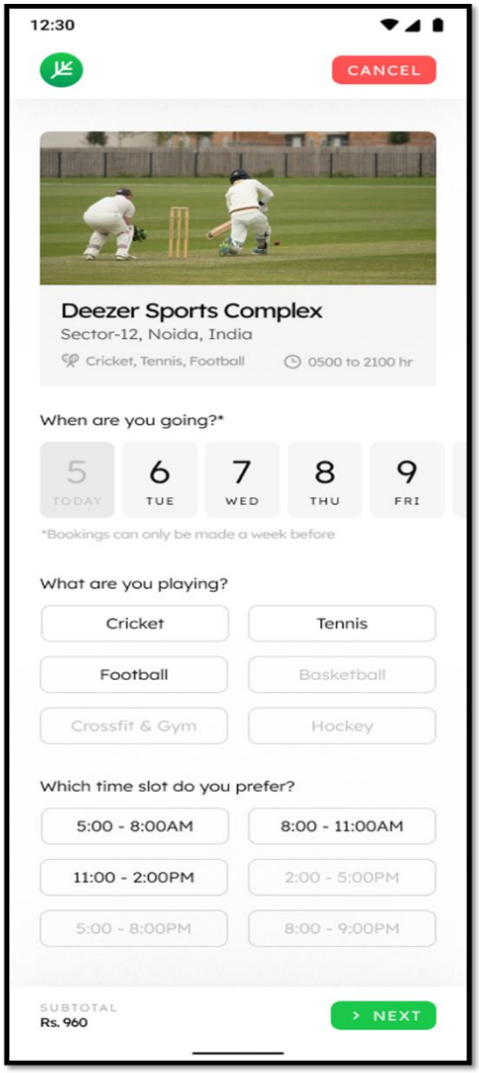
Home page



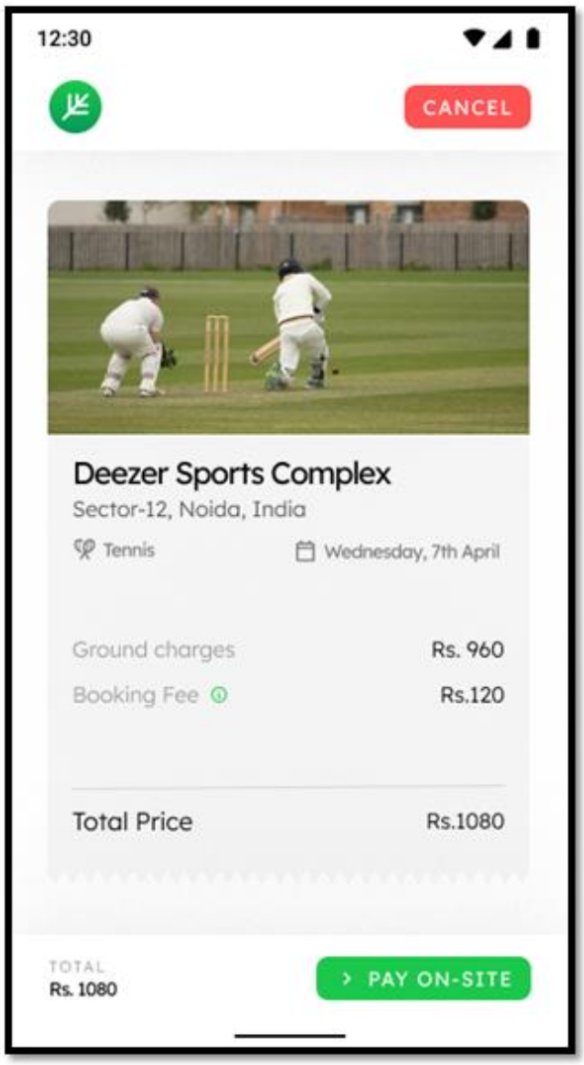
Home search page



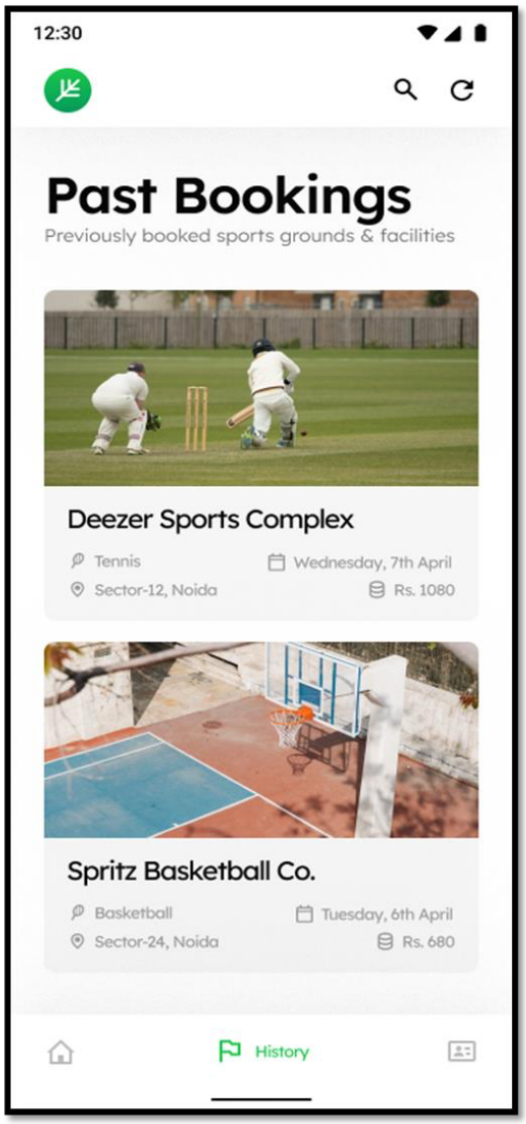
Booking Page

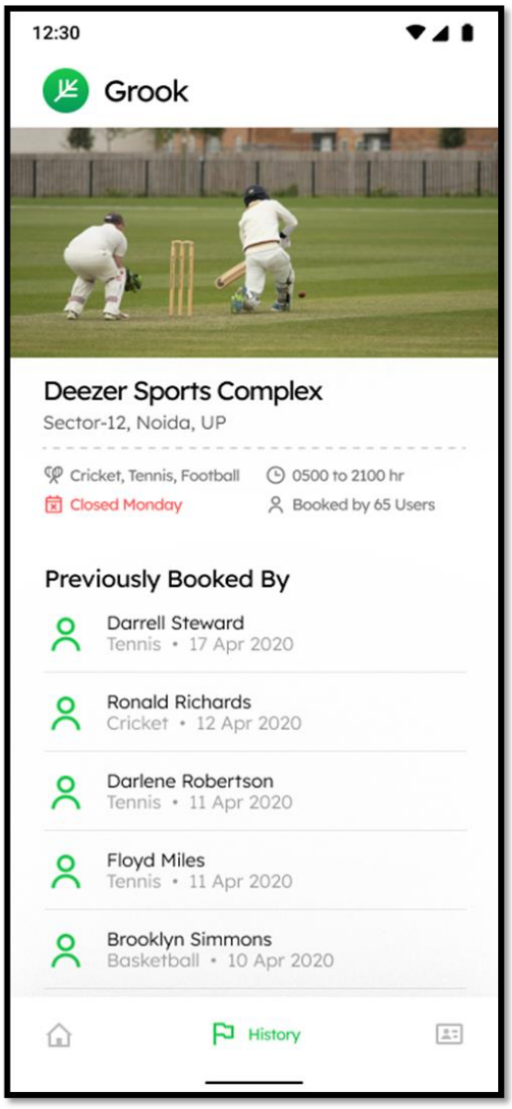


Booking Confirmation Page

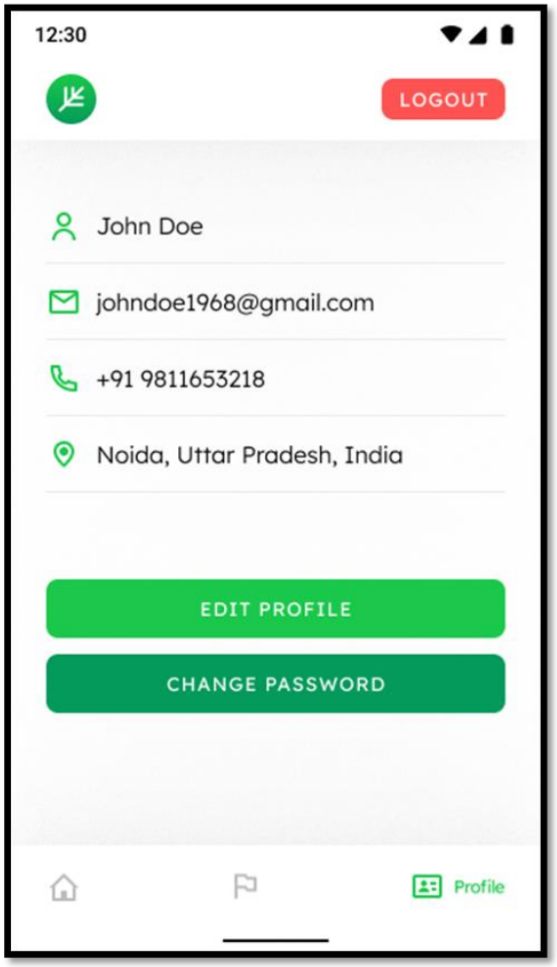


History Page



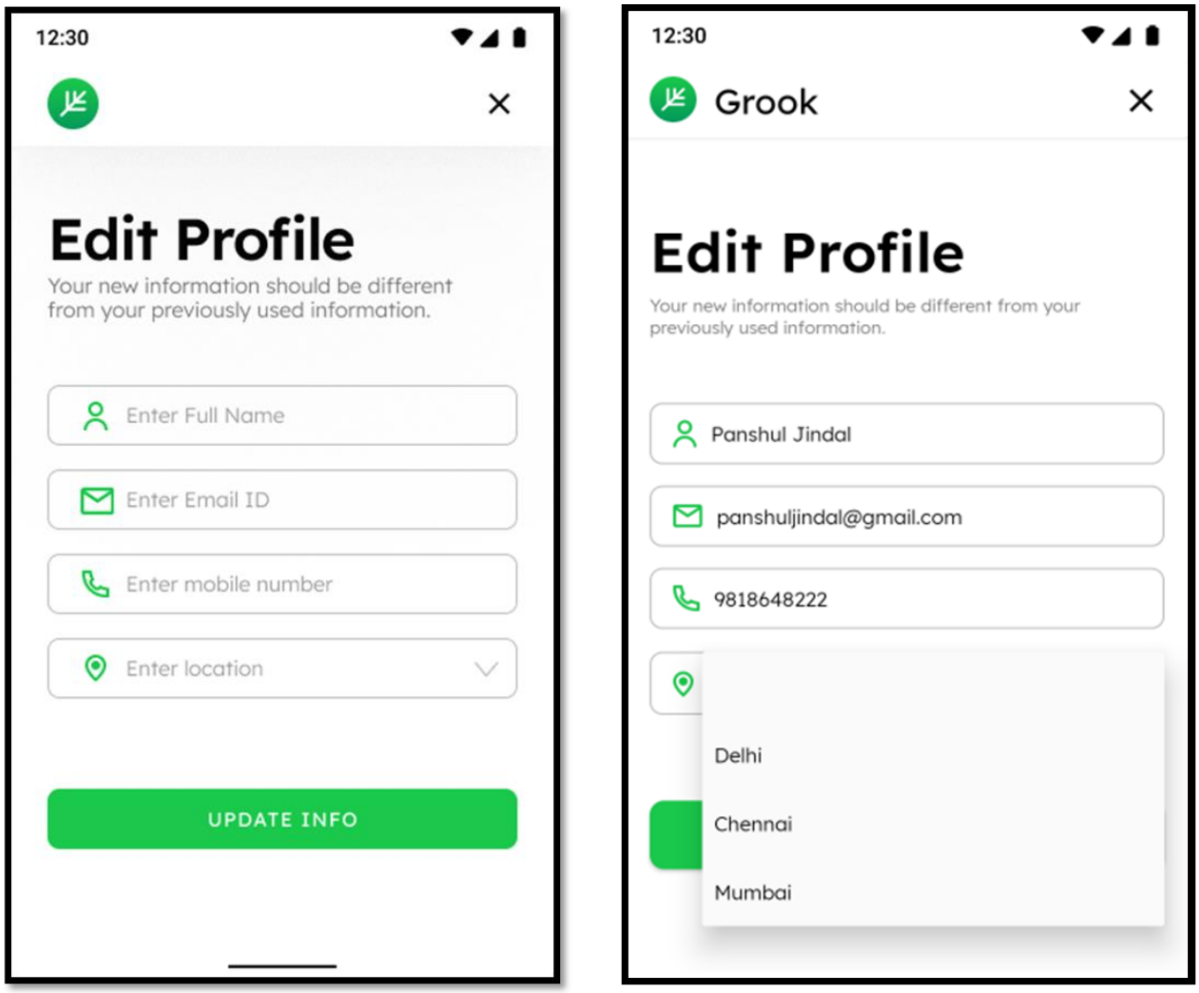


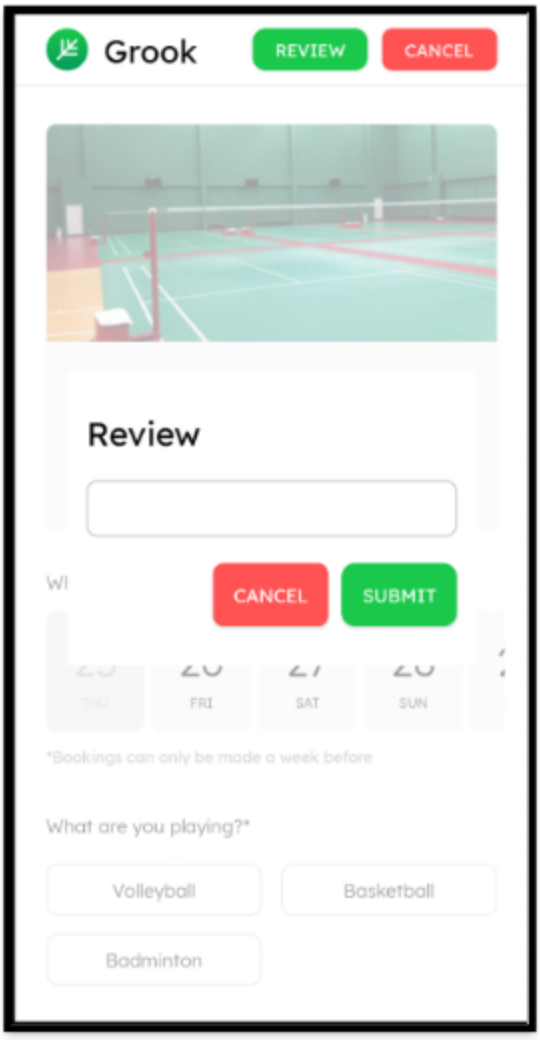
Profile Page



Reset Password

Edit Profile Page





Regression

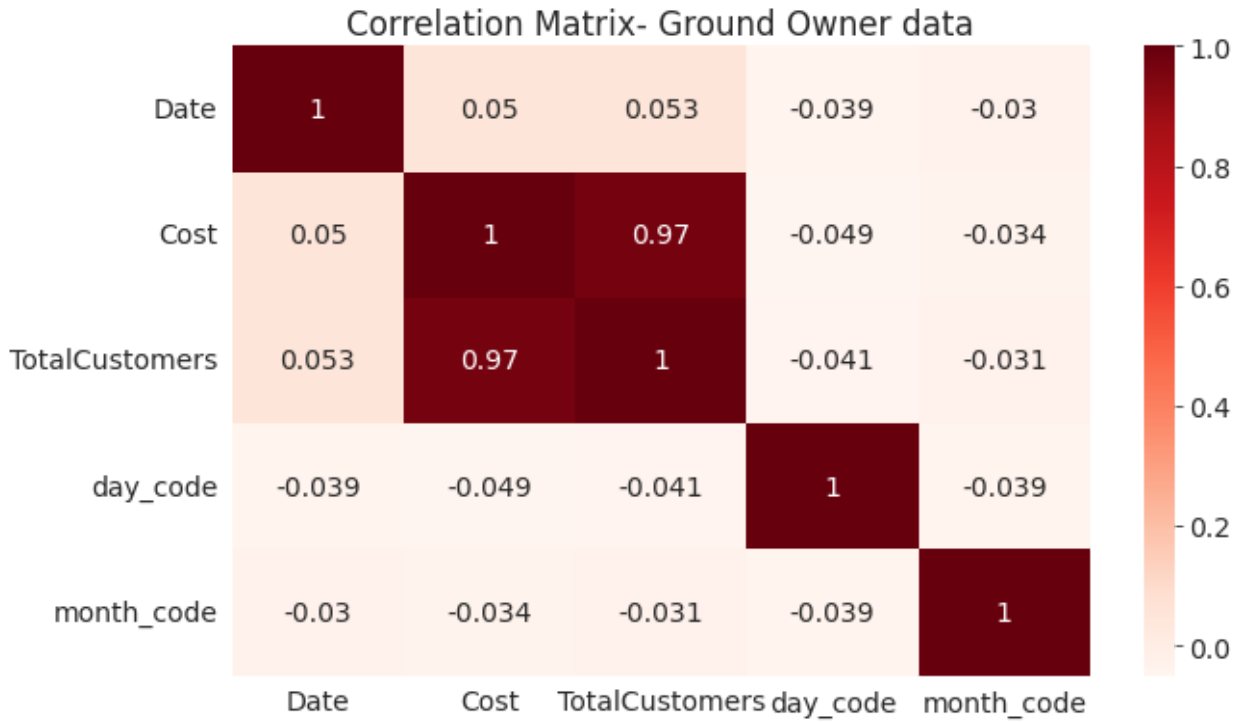
Different models will be constructed and tested for the data . Models include Linear ,polynomial , decision tree, ridge, lasso ,elastic net, random forest, .Our main focus will be to develop a hybrid model which chooses the best model for the input case.

Dataset for the models constructed includes feature columns as Date, Year, Day ,Month , GroundName , Cost , Total Customers. The dataset is split according to ground name to predict the review for the particular ground.

	Date	Year	Day	Month	GroundName	Cost	TotalCustomers
0	1	2010	Monday	August	Ground3	6400	27
1	2	2010	Saturday	May	Ground4	3200	13
2	3	2010	Thursday	July	Ground1	5000	20
3	4	2010	Thursday	February	Ground1	2700	12
4	5	2010	Wednesday	January	Ground3	6150	29
...	...	...	...	...	...	...	...
2994	2995	2021	Tuesday	August	Ground1	3400	14
2995	2996	2021	Friday	February	Ground2	2600	13
2996	2997	2021	Saturday	December	Ground1	2550	10
2997	2998	2021	Wednesday	January	Ground2	4550	22
2998	2999	2021	Tuesday	November	Ground3	2900	15

2999 rows × 7 columns

Correlation of features for the ground1 is calculated by the correlation matrix and represents the day and month as encoded values .



Our proposed model for voting combines multiple linear regression , polynomial regression , ridge regression , lasso regression , elastic net regression , decision tree regression , and support vector regressor. The validation accuracy of the voting model is 93%.

```
estimators
```

```
[('lr_2', LinearRegression()),
 ('lr_3', LinearRegression()),
 ('l4_ridge', Ridge(alpha=0.0001)),
 ('l5_lasso', Lasso(alpha=1)),
 ('l6_elasticNet', ElasticNet(alpha=0.005, l1_ratio=0.9)),
 ('regt1', DecisionTreeRegressor(criterion='mse', max_depth=5)),
 ('svr', SVR())]
```

```
from sklearn.ensemble import VotingRegressor
vr = VotingRegressor(estimators)
scores = cross_val_score(vr,X_train,y_train,scoring='r2',cv=10)
print("Voting regressor : ",np.round(np.mean(scores),2))
```

```
Voting regressor :  0.93
```

Our proposed model for Bagging, based on Hyperparameter tuning suggests that Lasso Regression model as base estimator,providing 100% of features to the estimators , providing 100% data from the dataset and including 20 estimators provides the best results of 94.5%.

```
estimators
```

```
[('lr_2', LinearRegression()),
 ('lr_3', LinearRegression()),
 ('l4_ridge', Ridge(alpha=0.0001)),
 ('l5_lasso', Lasso(alpha=1)),
 ('l6_elasticNet', ElasticNet(alpha=0.005, l1_ratio=0.9)),
 ('regt1', DecisionTreeRegressor(criterion='mse', max_depth=5)),
 ('svr', SVR())]
```

```
#finding which base estimator provides best results
n_samples = X_train.shape[0]
n_features = X_train.shape[1]

params = {
    'base_estimator' : [lr_2,lr_3,l4_ridge,l5_lasso,l6_elasticNet,regt1
    'n_estimators':[20,50,100],
    'max_samples':[0.5,1.0],
    'max_features':[0.5,1.0],
    'bootstrap':[True,False],
    'bootstrap_features':[True,False]
}

bagging_reg_grid = GridSearchCV(BaggingRegressor(random_state=1,n_jobs=
bagging_reg_grid.fit(X_train,y_train)
```

```
Fitting 3 folds for each of 336 candidates, totalling 1008 fits
```

```
Train r2 score : 0.950
Test r2 score: 0.945
Best r2 score through grid search : 0.945
Best parameters :  {'base_estimator': Lasso(alpha=1), 'bootstrap': Fal
se, 'bootstrap_features': False, 'max_features': 1.0, 'max_samples':
0.5, 'n_estimators': 20}
```

Our proposed model for XGBoost, based on Hyperparameter tuning, suggests that using decision trees as estimators with a learning rate of 0.3% , max depth of 2 for the tree and keeping the number of estimators as 20 provides the best results of 94%.

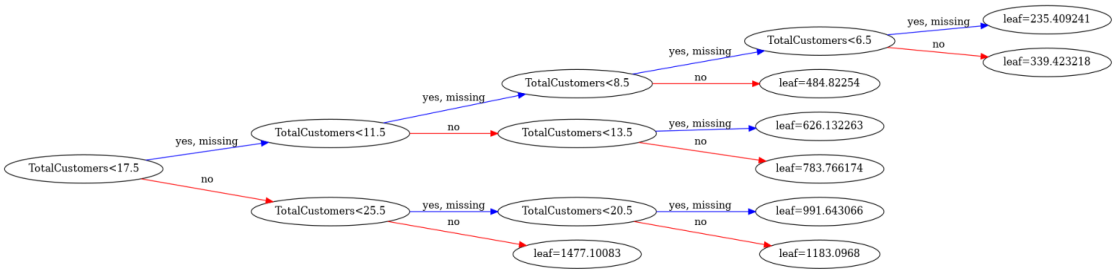
```
{'booster': 'gbtree', 'learning_rate': 0.3, 'max_depth': 2, 'n_estimat
ors': 20}
```

```
boosting_grid.best_score_
```

```
0.9406810945252729
```

	feature	importance
0	TotalCustomers	0.898575
3	Day_Sunday	0.008696
9	Month_February	0.008400
17	Month_September	0.007278
12	Month_June	0.007181
15	Month_November	0.007089
11	Month_July	0.006926
8	Month_December	0.006461
2	Day_Saturday	0.006074
7	Month_August	0.005915

Intermediate tree in the boosting regressor.



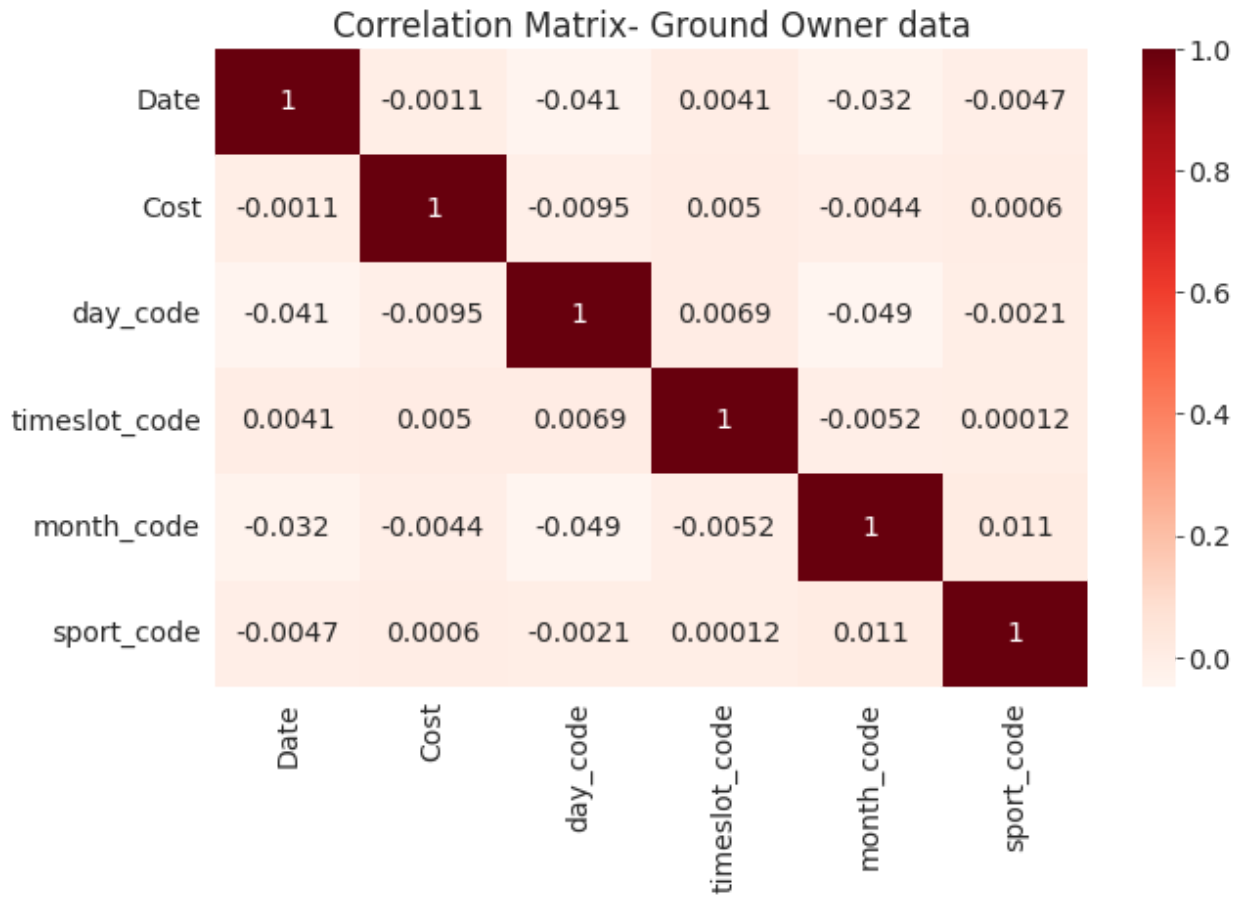
Classification

Different models will be constructed and tested for the data . Models include Linear , decision tree, neural net

	Date	Year	Day	Month	TimeSlot	Sport	GroundName	Cost
0	1	2010	Monday	August	timeslot3	sport5	Ground3	350
1	1	2010	Monday	August	timeslot1	sport3	Ground3	100
2	1	2010	Monday	August	timeslot6	sport6	Ground3	150
3	1	2010	Monday	August	timeslot4	sport3	Ground3	150
4	1	2010	Monday	August	timeslot3	sport5	Ground3	450
...	...	...	...	...	...	...	...	...
49595	2999	2021	Tuesday	November	timeslot2	sport5	Ground3	250
49596	2999	2021	Tuesday	November	timeslot4	sport6	Ground3	350
49597	2999	2021	Tuesday	November	timeslot5	sport4	Ground3	250
49598	2999	2021	Tuesday	November	timeslot2	sport4	Ground3	200
49599	2999	2021	Tuesday	November	timeslot7	sport6	Ground3	250

49600 rows × 8 columns

Correlation of features for the customer data of customers visiting ground1 is calculated by the correlation matrix and represents the day and month ,sport and timeslot as encoded values .



Our proposed Voting Classifier combining Logistic Regression , Random forest, K nearest Neighbors, Decision tree classifier for sport prediction gave a 0.16% accuracy when hard voting was performed. Hard voting involves choosing the output provided by the majority of the models. It gave an accuracy of 16% when soft voting was applied. Soft voting involves deciding the output based on the probabilities of the output given by the other models. The voting classifier gave an accuracy of 15% for timeslot prediction using hard voting and 14% using soft voting. Since the base models were providing an accuracy of less than 50% , using voting in this case of classification did not prove efficient.

For sport :

```
for estimator in estimators :
#     print(estimator[0])
    x = cross_val_score(estimator[1],X_ohe,y,cv=10,scoring='accuracy')
    print(estimator[0],np.round(np.mean(x),2))
```

```
logr 0.16
rf 0.16
knn 0.16
dt 0.16
```

```
from sklearn.ensemble import VotingClassifier
```

```
# hard voting :
voting_sport = VotingClassifier(estimators = estimators,voting="hard")
x = cross_val_score(voting_sport,X_ohe,y,cv=10,scoring='accuracy')
print(np.round(np.mean(x),2))
```

```
0.16
```

```
# soft voting :
voting_sport = VotingClassifier(estimators = estimators,voting="soft")
x = cross_val_score(voting_sport,X_ohe,y,cv=10,scoring='accuracy')
print(np.round(np.mean(x),2))
```

```
0.16
```

For timeslot :

```
logr 0.14
rf 0.14
knn 0.14
dt 0.14
0.15
0.14
```

Our proposed Bagging Classifier was trained using a decision tree as a base model and random forest as a base model . The decision tree bagging classifier provided 17% accuracy , random forest provided accuracy of 16% for sport prediction. The decision tree and random forest provided an accuracy of 15% and 14% for timeslot prediction.These models proved to be better than Voting classifiers.

For sport :

```
bagging_sport = BaggingClassifier(
    base_estimator = vt_clf1,
    n_estimators = 500,
    max_samples = 0.25,
    bootstrap = True,
    random_state=42
)
```

```
bagging_sport.fit(X_train,y_train)
```

```
BaggingClassifier(base_estimator=RandomForestClassifier(), max_samples=0.25,
                  n_estimators=500, random_state=42)
```

```
y_pred = bagging_sport.predict(X_test)
```

```
accuracy_score(y_test,y_pred)
```

```
0.1708185053380783
```

```
bagging_sport = BaggingClassifier(
    base_estimator = vt_clf2,
    n_estimators = 500,
    max_samples = 0.25,
    bootstrap = True,
    random_state=42
)
```

```
bagging_sport.fit(X_train,y_train)
```

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(max_depth=18,
                                                        random_state=42),
                  max_samples=0.25, n_estimators=500, random_state=42)
```

```
y_pred = bagging_sport.predict(X_test)
```

```
accuracy_score(y_test,y_pred)
```

```
0.16844602609727166
```

For timeslot :

```
bagging_timeslot = BaggingClassifier(
    base_estimator = vt_clf1,
    n_estimators = 50,
    max_samples = 0.25,
    bootstrap = True,
    random_state=42
)

bagging_timeslot.fit(X_train,y_train)

y_pred = bagging_timeslot.predict(X_test)
print(accuracy_score(y_test,y_pred))

bagging_timeslot.estimators_samples_ # which set of rows did the classifier get
```

```
0.15104784499802293
```

```
bagging_timeslot = BaggingClassifier(
    base_estimator = vt_clf2,
    n_estimators = 50,
    max_samples = 0.25,
    bootstrap = True,
    random_state=42
)

bagging_timeslot.fit(X_train,y_train)

y_pred = bagging_timeslot.predict(X_test)

print(accuracy_score(y_test,y_pred))

bagging_timeslot.estimators_samples_ # which set
```

```
0.14748912613681298
```

Our proposed stacking classifier including decision tree and random forest gave an accuracy of 17% for sport prediction and 14% for timeslot prediction, however the stacking classifier was weakened due to the weak decision tree and random forest models.

	Accuracy	MCC	F1
randf	0.295461	0.154537	0.294682
dtree	0.286067	0.143875	0.282398
stack	0.176505	0.010910	0.158277

	Accuracy	MCC	F1
randf	0.260358	0.137093	0.259155
dtree	0.255018	0.131042	0.252018
stack	0.144468	-0.000009	0.129350

Our proposed ADA Boosting model provided an accuracy of 15.6% for sport prediction and 14.5% for timeslot prediction,on performing hyperparameter tuning , the best model proposed involved 50 estimators, a learning rate of 0.001% and usage of the SAMME.R algorithm as the base estimator instead of SAMME algorithm , the accuracy for sport prediction increased to 16.9% and 16.9% for timeslot prediction.

For sport :

```
np.mean(cross_val_score(ada_sport,X_train,y_train,scoring='accuracy',cv=10))
```

```
0.1567310962808318
```

```
grid['n_estimators'] = [10,50,100,150]
grid['learning_rate'] = [0.0001,0.001,0.01,0.1,1]
grid['algorithm'] = ['SAMME','SAMME.R']


grid_search = GridSearchCV(estimator = ada_sport,param_grid = grid,n_jobs=-1,cv=10,scoring="accuracy")
grid_result = grid_search.fit(X_ohe,y)
```

```
grid_result.best_score_
```

```
0.16998830489818384
```

```
grid_result.best_params_
```

```
{'algorithm': 'SAMME.R', 'learning_rate': 0.001, 'n_estimators': 50}
```

For timeslot :

```
np.mean(cross_val_score(ada_timeslot,X_train,y_train,scoring='accuracy',cv=10))
```

```
0.1452619016901045
```

```
grid = dict()
grid['n_estimators'] = [10,50,100,150]
grid['learning_rate'] = [0.0001,0.001,0.01,0.1,1]
grid['algorithm'] = ['SAMME','SAMME.R']


grid_search = GridSearchCV(estimator = ada_timeslot,param_grid = grid,n_jobs=-1,cv=10,scoring="accuracy")
grid_result = grid_search.fit(X_ohe,y)


print(grid_result.best_score_)
print(grid_result.best_params_)
```

```
0.16998830489818384
{'algorithm': 'SAMME.R', 'learning_rate': 0.001, 'n_estimators': 50}
```

Sentiment Text Classification

Different models will be constructed and tested for the customer reviews. Models include Naive Bayes, Support Vector Machine, Feed forward Neural Network , LSTM(Long Short Term Memory) , GRU(Gated Recurrent Units). Our main focus will be to develop a hybrid model which uses different models and text representations to provide most relevant classification.

Preprocessing : Bag of words

dictionary: {0:"recurrent",1:"neural",2:"network",3:"artificial",4:"intelligence"}

example sentence: "artificial neural network and the recurrent neural network"

we don't remove the stopwords here since Recurrent neural networks are based on sequential data so here the sequence of all the words matter even the stopwords.

	artificial	**neural**	**network**	**and**	**the**	**recurrent**	**neural**	**network**	**final vector**
recurrent	0	0	0	0	0	1	0	0	1
neural	0	1	0	0	0	0	1	0	2
network	0	0	1	0	0	0	0	1	2
artificial	1	0	0	0	0		0	0	1
intelligence	0	0	0	0	0	0	0	0	0

final vector: {1,2,2,1,0}

The final vector is the one hot encoding of the sentence with respect to the vocabulary which is a dictionary containing a bag of words in the whole corpus.

Support Vector Machine

Methodology

1. takes in the vectors of the words
2. transforms it into a higher dimensional space and where the words can be compared to each other.
3. Finds the similar words using the kernel function .

4. Kernel function is derived by scaling the input points to a higher dimensional vector space and a model is formed that transforms the points to a higher dimensional vector space.
5. The points that are close to each other are given more priority and a hyperplane is formed which separates most of the points.
6. Training: It finds a decision boundary(hyperplane) and maximizes the margin between the boundary and the support vectors .
   support vectors are the vectors nearest to the boundary.
7. To train a Support Vector Machine Model the words of the tweets are represented as vectors. The vectors are in n dimensional space since a bag of word encoding is used.All the word vectors have the same dimension. Support Vector Machine moves the data into a higher dimensional space.
8. Now, it separates the data into groups using hyperplanes. Kernel function is used to build the hyperplane by finding support vectors in higher dimensions.
9. The support vectors are the vectors nearest to the hyperplane for a class. The best hyperplane is found using the cross validation which means the loss is calculated for each hyperplane and parameters are modified to find the best hyperplane separating the data.

Results of support vector machine classifier :

```
SVM :accuracy :  82.75


SVM :confusion_matrix:
 [[346  76]
 [ 62 316]]
precision: 84.80392156862744
recall: 81.99052132701422
NB :accuracy :  82.0
```

Naive Bayes

Methodology

Bayes Theorem

$P(A|B) = P(B|A).P(A)\}/P(B)$

For our case:

$P(y|X) = P(X|y).P(y)P(X)$

1. y : class label

2. X : feature vector

feature vector X :

$X = (x_{1},x_{2},x_{3},....,x_{n})$

Assume :

all the features are mutually independent

$P(y|X) = P(x_{1}|y).P(x_{2}|y)...P(x_{n}|y).P(y)/P(X)$

P(y|X) :posterior probability

$P(x_{i}|y)$: class conditional probability

P(y) : Prior probability of y

P(X) : Prior probability of X

select the class with the highest probability

$y=argmax_{y}P(y|X) ={argmax_{y}P(x_{1}|y).P(x_{2}|y)..P(x_{n}|y).P(y)}/{P(X)}$

ignore the denominator since it does not depend on the class label

$y = argmax_{y}P(x_{1}|y).P(x_{2}|y)..P(x_{n}|y)$

Since the probability is small , we can apply log to calculate the max.

argmax means that the argument y, x values will be passed for the entry in the dataset . x1 .. xn are the features and the y is the label for the entry in the dataset . The max value y is found for the data

$y = argmax_{y}log(P(x_{1}|y))+log(P(x_{2}|y))+..+log(P(x_{n}|y))+log(P(y))$

Prior Probability : P(y) : frequency

Class Conditional probability :$P(x_{i}|y)$

In the case of categorical variables, such as counts or labels, a multinomial distribution can be used

Results of Naive Bayes Classifier :

```
NB :accuracy :  82.0


NB :confusion_matrix:
 [[361  61]
 [ 83 295]]
precision: 81.30630630630631
recall: 85.54502369668246
```

Stacked Naive Bayes and SVM

Our Naive Bayes and SVM stacking model provided a training accuracy of 100% and a testing accuracy of 85% which is better than either of the models. The meta learner used in the stacking model was Logistic regression , this proves that the stacking models and assigning weights to them based on their prediction can lead to better results since each model has a particular subset of data which it predicts with higher accuracy.

Training Accuracies

	Accuracy	MCC	F1
nb	0.99375	0.987537	0.99375
svm	1.00000	1.000000	1.00000
stack	1.00000	1.000000	1.00000

Testing accuracies

	Accuracy	MCC	F1
nb	0.8200	0.638787	0.819589
svm	0.8275	0.655022	0.827613
stack	0.8500	0.699182	0.850020

Feed Forward Neural Network

Weights , biases and activation functions comprise most of the part of a neural network . The neural network is trained using the following methodology.

- Data Loading :
- Preprocessing:
- Train Test Split:
- Model Declaration
- Training Pipeline
    - Forward pass: prediction, loss
    - Backward pass: gradients
    - Update weights: gradient descent
- Validation and accuracy
- layer 1: x1 = W1 * X + b1

- layer 1(activation): h1 = Relu(x1)
- layer 2: x2 = W2 * h1 + b2log()
- output : p=sigma(x2)
- loss : -(ylog(p)+(1-y)log(1-p))
- gradient : d L(W1,b1,W2,b2)/ dW1 = (dL/dp) * (dp/dx2) * (dx2/dh1) * (dh1/dx1) * (dx1/dW1)
- optimizer : Parameter update :
- W1 =W1 - alpha (dL/dW1)

Training loss of Feed Forward Neural Network :

```
Epoch #1        Train Loss: 0.623
Epoch #2        Train Loss: 0.396
Epoch #3        Train Loss: 0.288
Epoch #4        Train Loss: 0.261
Epoch #5        Train Loss: 0.250
Epoch #6        Train Loss: 0.244
Epoch #7        Train Loss: 0.241
Epoch #8        Train Loss: 0.239
Epoch #9        Train Loss: 0.238
Epoch #10       Train Loss: 0.238
```

```
test_text = """
The ground maintainance was horrible and the staff was not cooperative
"""
predict_sentiment(test_text)
```

```
0.344: Negative sentiment
```

The Feed forward neural network , naive bayes classifier, and support vector machines do not consider the order of the words . The solution to that is Recurrent Neural Networks.

Word Embeddings visualization using LSTM.

Representation of the words

1. Bag of words , unique words using all words. one -hot encoding using the dictionary that you created. sentence as a vector with 1 and 0 for occurrence and non occurrence of the term in the sentence. Length of the encoding is the same as the length of the dictionary.
2. Integer encoding. assign a number to each unique word. for the sentence, the words will be represented according to the number

Cons:

- cannot portray the relationships between the words
- reason: basis in higher dimensional space, vectors are orthogonal, dot product is 0. No projection on any axis therefore they cannot show the relationships.
- The vectors are too sparse.
  Solution:

3. Word Embedding New space, where the words are transformed to , it is a hyperparameter of the model like the number of hidden layers in a neural network.

Dot products can be taken and the relations between words can be represented.

strong correlation of words. The model takes words,puts through the embedding layers., good or bad review,matches the training label.and then back propagates through the model and changes parameters.

Model now can predict positive and negative and can also show a correlation between words.

GRU: Gating Recurrent Units

Recurrent neural networks feed the output to itself , and can be done using a loop.
The type of rnn to be used:many to one

Traditional rnn has the vanishing gradient problem: the words appearing in the beginning start to lose their meaning in the final computation as more words are passed into the recurrent neural network.

RNN architecture used : GRU (Gated Recurrent Unit) uses update and reset gates to decide what is important to keep in the sentence uses different activation functions

1. inputs: takes in current state(xt) and the previous state inputs (h(t-1))
2. reset gate: rt = sigma(Wr * xt + Ur * h(t-1) +br
   br = reset bias
   trainable parameters: Wr, br
3. update gate: zt = sigma(Wz * xt + Uz * h(t-1) +bz)
   trainable parameters: Wz, bz
   bz = update bias
4. hidden state : ht = (1-zt) o h(t-1) + zt o (Wh*xt + Uh(rt o h(t-1))+ bh)
5. The trainable parameters include: Wh, bh

Train loss of Gated recurrent unit neural network:

```
Epoch #1        Train Loss : 0.693
Epoch #2        Train Loss : 0.652
Epoch #3        Train Loss : 0.489
Epoch #4        Train Loss : 0.400
Epoch #5        Train Loss : 0.358
Epoch #6        Train Loss : 0.336
Epoch #7        Train Loss : 0.322
Epoch #8        Train Loss : 0.312
Epoch #9        Train Loss : 0.305
Epoch #10       Train Loss : 0.298
```

```
test_text = """
The ground maintainance was horrible and the staff was not cooperative
"""
predict_sentimentGRU(test_text)
```

```
0.0811: Negative sentiment
```

Conclusion

In this project, we developed an android application which was deployed on google play store. The application provided functionalities for the user to find sports facilities in their vicinity, book convenient time slots and also form teams with other users with similar sports interests.We used the structure of the booking data of the application and trained regression and classification models including multiple linear regression , polynomial regression, decision trees and applied ensemble techniques including stacking , bagging , boosting, regularization and hyperparameter tuning to predict total revenue of a ground, most likely sport to be chosen by the customer, most likely time slot chosen by the customer . We also trained SVM, NB , Stacked SVM and NB, Feed Forward Neural Network and Gated Recurrent Unit Neural Network models to predict the sentiment of customer reviews of a ground.In the future the objective would be to integrate the machine learning models in the application and provide the user with real time feedback. We would also like to try and implement bagging ensemble models for sentiment classification using deep learning models.

Link to Code and Deployed Application

ML Code : https://github.com/NeelChoksi/sem6_TARP_RP
Grook Application: https://play.google.com/store/apps/details?id=com.panshul.grook

References

[1] K. Rathan, S. V. Sai and T. S. Manikanta, "Crypto-Currency price prediction using Decision Tree and Regression techniques," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 190-194, doi: 10.1109/ICOEI.2019.8862585.

[2] V. Subramaniyaswamy, M. V. Vaibhav, R. V. Prasad and R. Logesh, "Predicting movie box office success using multiple regression and SVM," 2017 International Conference on Intelligent Sustainable Systems (ICISS), 2017, pp. 182-186, doi: 10.1109/ISS1.2017.8389394.

[3] Alsmadi, Izzat & Alhami, Ikdam. (2015). Clustering and Classification of Email Contents. Journal of King Saud University - Computer and Information Sciences. 12. 10.1016/j.jksuci.2014.03.014.https://www.researchgate.net/publication/270594957_Clustering_and_Classification_of_Email_Contents

[4]Bhowmick, Alexy & Hazarika, Shyamanta. (2018). E-Mail Spam Filtering: A Review of Techniques and Trends. 10.1007/978-981-10-4765-7_61.https://www.researchgate.net/publication/320703241_E-Mail_Spam_Filtering_A_Review_of_Techniques_and_Trends

[5]Rathod, S. B., & Pattewar, T. M. (2015). Content based spam detection in email using Bayesian classifier.https://www.academia.edu/27150903/Content_Based_Spam_Detection_in_Email_using_Bayesian_Classifier

[6] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12. https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications

[7]M. U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," in IEEE Access, vol. 8, pp. 58080-58093, 2020, doi: 10.1109/ACCESS.2020.2982538. https://ieeexplore.ieee.org/document/9044300

[8]Karandeep Singh Talwar,Abhishek Oraganti,Ninad Mahajan,Pravin Narsale.Recommendation System using Apriori Algorithm, 2015.IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 01, 2015 | ISSN (online): 2321-0613.

[9]Pandya, Sharnil & Shah, Jaimeel & Joshi, N. & Ghayvat, Hemant & Mukhopadhyay, S.C. & Yap, Moi Hoon. (2016). A novel hybrid based recommendation system based on clustering and association mining. 1-6. 10.1109/ICSensT.2016.7796287.

[10]Feng, W., Zhu, Q., Zhuang, J. et al. An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. Cluster Comput 22, 7401–7412 (2019). https://doi.org/10.1007/s10586-017-1576-y

[11]Nguyen, Dong & Smith, Noah & Rosé, Carolyn. (2011). Author Age Prediction from Text using Linear Regression. 115-123. https://aclanthology.org/W11-1515.pdf

[12]M. Kavitha, G. Gnaneswar, R. Dinesh, Y. R. Sai and R. S. Suraj, "Heart Disease Prediction using Hybrid machine Learning Model," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1329-1333, doi: 10.1109/ICICT50816.2021.9358597.

[13]I. Ahmad, M. Basheri, M. J. Iqbal and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," in IEEE Access, vol. 6, pp. 33789-33795, 2018, doi: 10.1109/ACCESS.2018.2841987.

[14]X. Wang, X. Wang, B. Ma, Q. Li and Y. -Q. Shi, "High Precision Error Prediction Algorithm Based on Ridge Regression Predictor for Reversible Data Hiding," in IEEE Signal Processing Letters, vol. 28, pp. 1125-1129, 2021, doi: 10.1109/LSP.2021.3080181.

[15]K. Talele, A. Shirsat, T. Uplenchwar and K. Tuckley, "Facial expression recognition using general regression neural network," 2016 IEEE Bombay Section Symposium (IBSS), 2016, pp. 1-6, doi: 10.1109/IBSS.2016.7940203.