Optimised method to identify multiservice sports platform using ML : Grook

Khushee Jain, UG Scholar Neel Choksi, UG Scholar Panshul Jindal, , UG Scholar

SCOPE, Vellore Institute Of Technology, Vellore

Professor Amutha Prabhakar M.

1. Abstract

Grook provides an app-based solution to book a ground in a particular time slot . It allows aggregation of all available sports facilities in the vicinity of the user's location.

Ensemble Machine Learning Techniques in Regression and Classification are suggested to enhance the existing algorithms. Sentiment classification models including SVM , Naive Bayes , RNN , LSTM , GRU , Feed forward neural networks are enhanced using stacking techniques. Regression algorithms including Linear , Ridge and Lasso Regression are enhanced using Random Forest and Boosting Techniques.

Stacking , Boosting , Bagging , Regression , Sentiment classification.

2. Introduction

2.1About the application

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.

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.

2.2 Data to be collected from application

User data including customer reviews for a ground can be collected and tested against the trained model for finding out the sentiment of a text sentence. Data points including timeslot , sport chosen ,cost , day , month , year of visit and ground visits are collected and analyzed and frequency of users visiting for a sport , or in a timeslot or on a particular day is predicted using regression models. Revenue of a ground is predicted based on previous bookings on that ground.

3. Related Work

3.1 Learnings from the sentiment base paper [1]

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.

3.2 Learnings from the regression base paper [2]

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 algorithms, 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.

4. Methodology

4.1 Preprocessing

The input text to the deep learning model can be represented in various forms including Bag Of Words, Word Embedding and TFIDF representation . Before converting the text into the desired form , we need to perform some preprocessing on the text data which is in the form of sentences. First step in preprocessing is tokenization where the words are individually represented in a list. Following tokenization , we can perform stemming or lemmatization which reduces the word into the root form so that different forms of words expressing the same feeling are considered as the same word.

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}

Table 1. Bag Of Words Representation.

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. Instead of representing the 1 or 0 in the cell we can replace it with the TFIDF value which is a better representation of the words relative to other words . TF indicates the Term Frequency which is the frequency of term t in a review d. IDF is the Inverse Document Frequency , found out by taking the log of ratio of total reviews in the dataset and dividing it by the number of reviews wherein the term appears.

4.2 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.

4.3 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

4.4 Feed Forward Neural Network based on bag of words

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)

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.

4.5 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.

4.6 RNN LSTM GRU

The word embedding representation of the words can be fed into the Recurrent neural network . It reads the sentence word by word like a human being by having an input , output and a hidden state in the base unit of RNN. It is a sequential memory therefore it suffers from the problem of Vanishing gradient. It forgets the words read by initial units as the RNN progresses forward to read the whole statement. A solution to the RNN vanishing gradient Problem is the use of LSTM(Long Short Term Memory) and GRU(Gated Recurrent) networks.

LSTM and GRU have gates that determine whether to keep or discard the word in the memory. Hidden states are passed forward similar to RNN . GRU has only a reset gate and an update gate. The update gate either adds or forgets the word. These gates add or remove information from a cell state that acts as a long term memory for the Network hence they solve the vanishing gradient problem in the RNN.

4.7 Stacking

Models stacked here are the support vector machine and naive bayes. Another stacking is done using the deep learning models involving feed forward neural network using bag of words representation , feed forward neural network using the word embeddings representation and RNN,LSTM, GRU model .

Stacking is an Ensemble machine learning algorithm where multiple models are used to train from the dataset . The models are provided with the inputs and the outputs of all the models are fed into an meta learner model which has the job of determining the final output based on the outputs from the base models . Logistic regression is often used as a meta learner. The meta learner assigns weights to the base models based on their performance on data points. Stacking is used in this case since the models provide better results for particular data points.

4.8 Linear Lasso Ridge

Linear Regression model uses the least square sums and fits a line to the data points in the x-y plane. This line fitting suffers from overfitting , although it provides minimum variance . But to improve the result of the model , regularization techniques including Lasso Regression and Ridge Regression are used . These techniques add a little bias into the model so that the line does not fit exactly to the points in the training dataset. Ridge Regression adds lambda*$(slope)^2$ as variance and Lasso Regression adds lambda*|slope| as bias. The lasso regression model has less variance than the Ridge regression model.

4.9 Linear Regression

Linear regression is a statistical regression method that illustrates the relationship between continuous variables and is used for predictive analysis. Simple linear regression is used to model a single input variable, whereas multiple linear regression is used to model many input variables. The relationship between the input variables used for the predictive analysis is represented by a slanted straight line generated by this model. Mean Squared Error cost function is used which is calculated by taking the average of the squared error that occurs between the predicted and actual values. This cost function optimises the regression coefficients or weights and assesses the performance of a linear regression model. An important application is to determine how much precise the mapping function is that converts an input variable to an output variable.Gradient descent is a method of updating line coefficients by decreasing the cost function by a random selection of coefficient values and then iteratively updating the values to attain the least cost function. MLR predicts one or more continuous variables based on other data set qualities, recognising existing connections among variables, and is commonly used in multivariate settings. Perform a correlation study to verify the relationship between the various crowd-sourced ratings. The regression model seeks to establish whether there is a linear relationship between customer rating taken as dependent variable and customer sentiment polarity taken as independent variable when making predictions based on customer rating and customer sentiment polarity. The model identifies that there is any linear relationship between customer sentiment polarity and customer rating.

4.10 Lasso Regression

Least Absolute Shrinkage and Selection Operator is the acronym for Lasso.

It is a statistical formula for regularizing data models and selecting features. Regularizing is an important idea to avoid overload of data,

especially if the data read and examined varies widely. Regularizing is done by adding the word “penalty” to the best equations produced from trained data to achieve low variance of the tested data, as well as pressing the coefficients of the predictive variables to limit their impact on the output variability. Lasso regression is a common type. For a more accurate prediction, it is preferred over retrospect. Shrinkage is used in this model. Data values are reduced to a central point known as mean in shrinkage. Simple, short models (i.e. models with fewer parameters) are encouraged by the lasso process. This type of retrospective is suitable for models with high multicollinearity or if you wish to automate the model selection process, such as dynamic selection and parameter removal. The standard L1 method is used in Lasso Regression. Used when we have a large number of features because it makes feature selection automatically. The regularity of the L1 adds a limited charge of the total value of the coefficient. This type of adjustment can lead to models with a few coefficient levels. Some coefficients may be 0, and the model will be excluded. When the charge becomes large, the coefficient is close to zero (suitable for producing simpler models).

4.11 Ridge Regression

There are basically two situations in which ridge regression is useful. The first is to estimate the regression parameters via the posterior mean, assuming a multivariate normal spherical prior distribution with variance parameter. The second is to estimate the regression parameters via the posterior mean, assuming a multivariate normal spherical prior distribution with variance parameter.

Restricted least squares formulation is the second. To estimate the regression parameters subject to a spherical limitation of the kind pfl = constant, we apply the Lagrange multipliers approach. Alternatively, the problem can be seen as a minimization problem with the constraint of being on a given sum of squares contour.Variations involving ellipsoidal rather than spherical distributions and limitations are not fundamentally different in either instance, but they do require a parameter translation and a change in the definition of mean squared error. When one has prior knowledge of the relationships between the indicators, ridge regression may be used. Ridge regression should not be employed in any other context. Thus, a mechanical application of ridge regression "to improve the conditioning" or any other such ambiguous "improvement" must be condemned outright. Ridge regression isn't a silver bullet. It's a method for incorporating or assuming past and/or practical information of a given type.Furthermore, simulations impose certain extremely strict constraints, skewing the results in favour of the ridge technique. Furthermore, whether estimated using least squares or ridge regression methods, these modest values often result in nonsignificant regressions with no practical interpretive value. The broad conclusion that ridge regression is "always" better than least squares is frequently unfounded.

4.12 Random Forest(Bagging) , Boosting

Random forest is a bagging model where the same model is fitted with different datasets . The initial dataset can be divided randomly into sub datasets and fed into the same model . Repetition of data points occurs.

Extreme Gradient boosting is a boosting ensemble technique where the data is fed into an instance of the same model . Then the results are passed into the second instance. The data passed to the instances indicate the points where the previous instance failed. The current instance keeps those points and makes predictions, it also makes some mistakes and passes that data to the next model. This way we can smoothen the decision boundary of the model.

4.13 Bagging: Random Forest

Random forest regression is a supervised machine learning technique that is particularly useful when other machine learning techniques appear to be computationally expensive. Here comes the notion of ensemble learning, which is a strategy for making more accurate predictions by combining predictions

from various machine learning algorithms or predictions from the same algorithm numerous times. Predictions from a single individual model may not be as accurate as desired, thus numerous models, such as three K&N support vector machines, might be developed on the same data to obtain combined predictions. Similarly, we can aggregate the predictions of numerous decision trees to get a final combined prediction. There are two types of symbol learning: The first is that trees are computationally expensive. The second is that they are very sensitive to the data on which they are trained, resulting in deviations in predictions if the underlying data is changed. Random forest solves this problem. It is a supervised learning technique that uses ensemble learning to perform both a regression and classification task. It constructs multiple decision trees over the course of a training time and combines the results of predictions from each decision tree. Because the trees in the random forest only operate in parallel, let's look at how the algorithm constructs a random forest number.The first step is to select K data points at random from the training set, so we are using the entire data set and then selecting a subset or K data points from it. The second step is to create a decision tree based on those K data points. As part of this stage, you will create a decision tree based on that subset of observations or K data points. The third step is to select the number and type of trees you want to construct, then repeat steps one and two. Now, as part of this phase, you will select the number and type of trees you want to construct.You want to build a lot of regression decision trees, so you repeat steps one and two over and over again. step four for a new data point, make each of your M three trees predict the value of y for the data point and assign the new data point the average of all of the predicted Y values as a part of this step you get multiple predictions from these decision trees so basically we take all of these decision trees to get the predictions that is each of them predicting the value of target variable Y for the new data point for whom we want to get prediction algorithm assigns the average value of all the decision trees predicted Y values to this new data point, so if the number of trees is set to a thousand, you'll get individual predictions for the value of y for each of the thousand trees, and then we'll take the average of the Y values of all the thousand trees, which is why it's called a random forest. Another issue to consider is whether or not your prediction accuracy improves as a result of the forecasts of some of the ideal individual decision trees being balanced out by the predictions of the other not so perfect decision trees.Because any change in the data set is less likely to impact the forest of these decision trees, these ensemble arabic algorithms are more stable. Please note that each decision tree draws a random sample from the original data set when generating its split, adding a further element of randomness that prevents overfitting.

4.14 XG Boosting

Gradient boosting is a technique for resolving regression and classification problems. This is a multiple-iteration sequential ensemble learning algorithm. The optimization framework for absolute differentiable loss functions has been modelled. Loss function, weak learner, and additive model are the parameters utilized in the gradient boosting algorithm. XGBoost is a scalable tree boosting system that is frequently used by data scientists. It is capable of solving real-world size problems with little resources. Decision tree-based algorithms are now considered best among the rest for small-to-medium data whether it is structured or tabular. XG Boost can be used in a variety of situations.Regression, classification, ranking, and user-defined prediction issues can all be solved using it. Supports all major programming languages, including C++, Python, R, Java, Scala, and Julia, and is portable to many operating systems.It also works with Flink, Spark, and other ecosystems and enables cloud integration such as AWS, Azure, and Yarn clusters. To avoid overfitting, it penalises more complex models using both LASSO and Ridge regularisation.XGBoost naturally accepts sparse input features by finding the optimum missing value based on training loss. To successfully determine the ideal split points among weighted datasets, it uses the distributed weighted Quantile Sketch algorithm.

5. Experiments and Results

Jupyter notebooks are hosted in the github repository :
https://github.com/NeelChoksi/sem6_TARP_RP/blob/main/Regression_final.ipynb
https://github.com/NeelChoksi/sem6_TARP_RP/blob/main/Sentiment_final.ipynb

Predicting the total revenue for the ground using regression.

Dataset

The Customer data is collected according to the customers visiting the ground on a particular date. The data is generated randomly using inbuilt libraries in python. The Dataset consists of the Date, Year , Day , Month , GroundName, Cost and Total Customers visiting the ground on that day.The data frame is depicted in Figure.1.There are 2999 entries for 4 grounds from 2010 to 2021
Each customer selects a sport in a timeslot using the application and pays a cost associated with it. The data for every customer is shown including the date, year, day , month , timeslot , sport , ground name and the cost paid. The data frame is depicted in Figure.2. There are a total of 49600 rows and is decomposed for the dataset shown in figure 1.

	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

Figure 1. Customers visiting the ground according to the day.

	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

Figure 2. Customer sport and timeslot choice selection according to transaction.

Analytics

Dataset is constructed using a random function in python . A data point on the booking dataset consists of the Date which is treated as a serial number, Day , Month , GroundName, Cost and total customers visiting on that day and the revenue generated by the ground owner on that day. Various features of the dataset are used to show the behavior of the data and perform feature engineering.
Revenue per day distribution of a ground owner is shown using a histogram in Figure.3 . Revenue of a day is shown for the ground owner according to year, month , day in Figure.4 . Number of people visiting the ground according to year,month,day is shown using a histogram in Figure.5 . Relationship of Day and Month with cost is shown using a violin plot. Correlation among the variables is shown using a correlation matrix in Figure.6.

The revenue from the customers visiting the ground in a timeslot is shown in Figure.7. The revenue from the customers visiting the ground choosing a sport is shown in Figure.8.

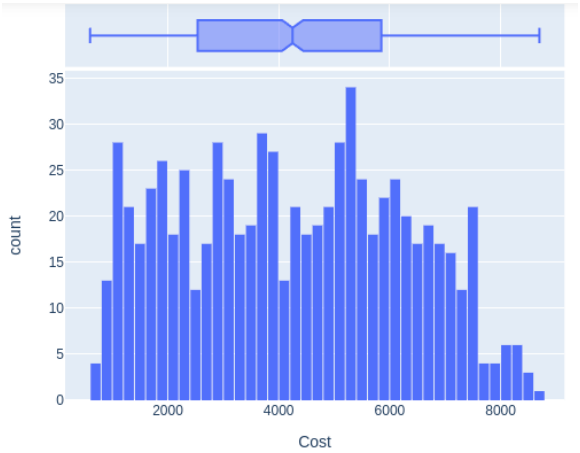


Figure 3. Revenue per day distribution of the owner of Ground 1.

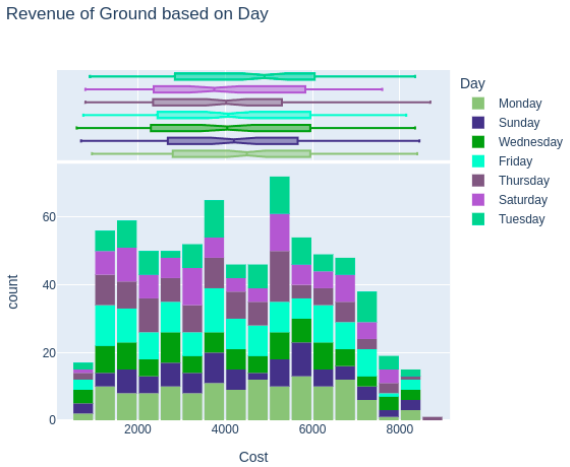
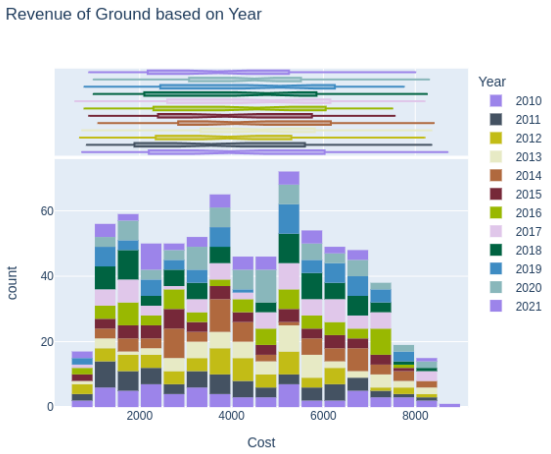
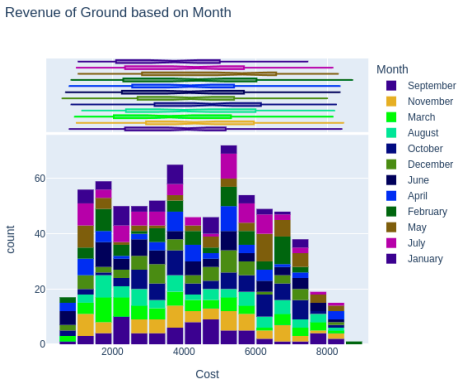
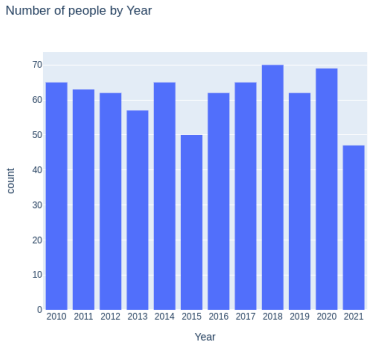
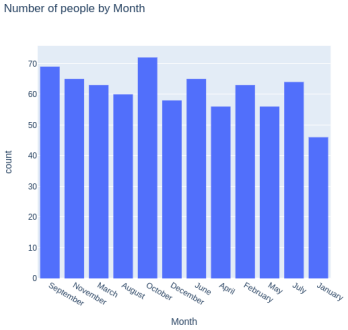


Figure 4. Yearly,monthly and daily revenue of the owner of Ground 1.









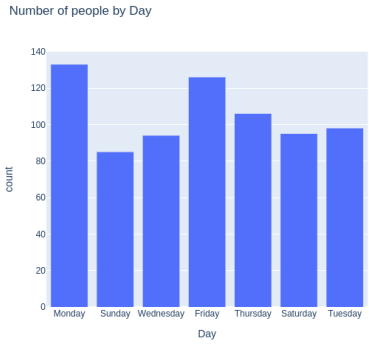


Figure 5. Distribution of number of customers according to Day,Month,Year of the owner of Ground 1.

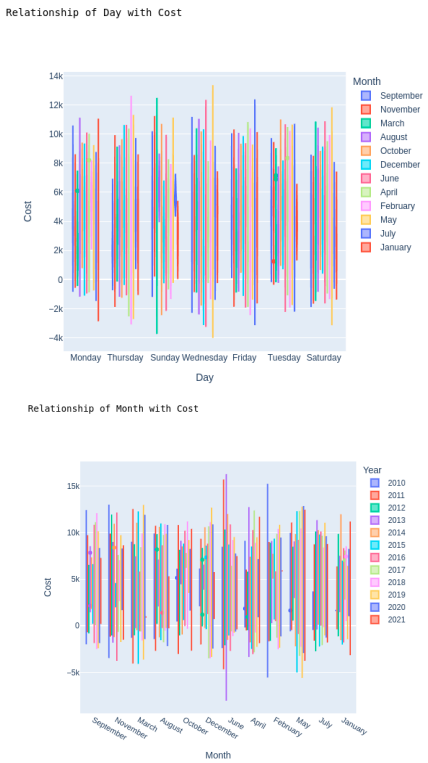


Figure 6. Relationship of revenue with the Day and Month of the owner of Ground 1.

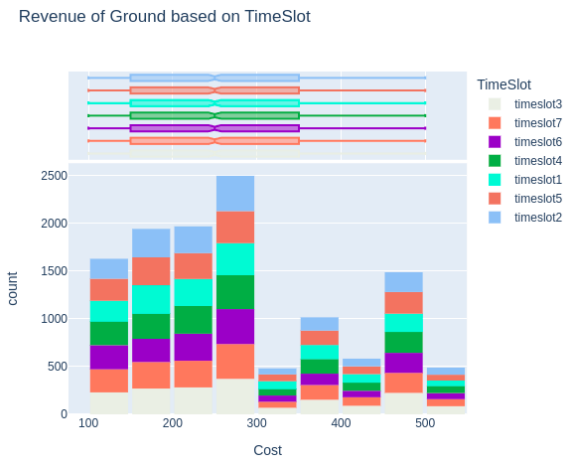


Figure 7. Revenue from customers in a timeslot of the owner of Ground 1.

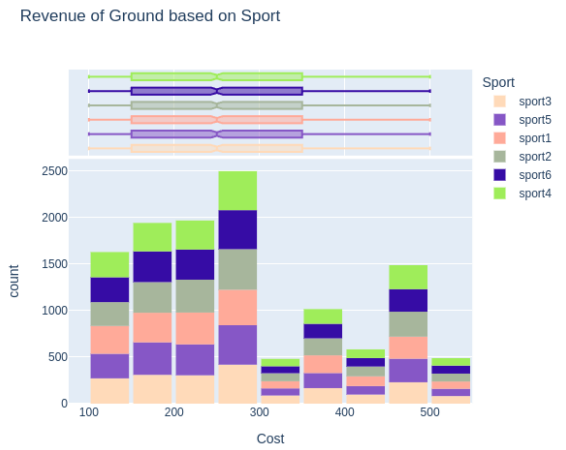


Figure 8. Revenue from customers choosing a sport of the owner of Ground 1.

Predicting Total Revenue of the day for the Ground using regression.

In the base paper [2] the authors have combined Decision Trees and Random forest .

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%.

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%.

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%.

A simple linear regression model can depict the relationship between number of customers visiting the ground on that day against the revenue generated on that day. The points are plotted in Figure 9. The regression line can be trained which will fit the data by training the weight and bias in the equation revenue = weight * totalCustomers + bias. The regression line is shown in Figure 9. The model minimizes the distance from the points to the line and adjusts the weights and biases for the best fit line The validation accuracy of the model is 94.5%. Our Dataset contains of more than one columns as feature set therefore to predict the Total Cost of the day , we need to use Multiple linear regression by training the model based on the equation : revenues = weight1*totalCustomers + weight2*Day + weight3*Month . One hot encoding is used to represent the Day and Month as a vector of length 7 and 12 with value 1 in the feature existing in the data point and 0 for the rest of the columns. Validation accuracy of the multiple linear regression model is 94.4%.

Polynomial Linear Regression is used by transforming the weights into higher dimensional equations .Validation accuracy of the model is 93.5%.

The linear regression model might overfit the training data , hence there is a need for regularization of the models and the method used is Ridge Regression which is an L2 regularization method , it adds a bias lambda to the training function to prevent the model from overfitting the training data . Ridge regression should be used when it is certain that all the features are important as the lambda it adds does not allow the model to eliminate any features whereas Lasso Regression should be used when only some of the features are important in the dataset, it also adds a bias to the training equation . ElasticNet regression can be used when the importance of the features is unknown, it adds biases of Lasso and Ridge Regression to the model for training. The Ridge Regression model gave a validation accuracy of 93.9%, Lasso Regression model gave a validation accuracy of 94% and 95%, hyperparameter tuning for different values of lambda for lasso regression was done. ElasticNet regression gave a validation accuracy of 94.4%.

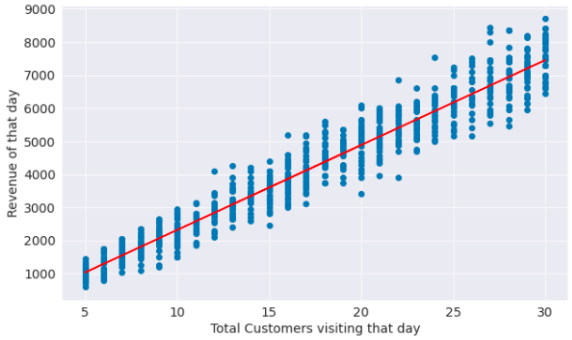


Figure 9. Simple Linear Regression.

Decision Tree Regression

Decision Trees are generally used for solving classification problem , but here we used it for solving a regression problem where the total cost was predicted with a validation accuracy of 93.6%.Hyperparameter tuning of the model was done to find out the best set of parameters shown in Figure 10.

```
GridSearchCV(estimator=DecisionTreeRegressor(),
             param_grid={'criterion': ['mse', 'mae'],
                         'max_depth': [2, 4, 6, 8, 10],
                         'max_features': [0.25, 0.5, 1.0],
                         'min_samples_split': [0.25, 0.5, 1.0]})
```

Figure 10. Hyperparameter Tuning of Regression Tree

Voting

Voting is based on the ideology of knowledge of the crowd. Multiple models are combined and the prediction from each model is averaged to give the final prediction . The models used in the voting model include multiple linear regression , polynomial regression , ridge regression , lasso regression , elastic net regression , decision tree regression , support vector regressor. The validation accuracy of the voting model is 93%.

Bagging

Bagging is an ensemble technique where bootstrap aggregation is performed. Multiple base models are trained with different subsets of the dataset . We have done hyperparameter tuning to find out which model provides the best results when used as a base model , along with other parameters including number of models in the bagging regressor, maximum sample size provided to a base model. The best results of 94.5% validation accuracy were provided when the bagging regressor was trained with the parameters shown in Figure 11.

```
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}
```

Figure 11. Hyperparameter Tuning of Regression Tree

XGBoost

Extreme gradient boosting uses various decision trees to predict the values and train on the wrong predictions . The errors of the previous model are learned by the current model also by using decision trees.One of the intermediate trees is shown in Figure 13. . We got a validation accuracy of 94%, cross validation was also done. The hyper parameters of the model were tuned and the best model gave the validation accuracy of 94.06% when using the parameters shown in Figure 12. Importance of each feature is found out , the top 10 important features are shown in Figure 14.

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

Figure 12. Hyperparameter Tuning of Regression Tree

Figure 13. Intermediate tree in XGBoost

	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

Figure 14. Feature Importance

Predicting the time slot, sport for the ground using classification

Our proposed Voting 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.
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.

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 as described in paper [24], the accuracy for sport prediction increased to 16.9% and 16.9% for timeslot prediction.

Decision Tree Classifier

The decision tree classifier is used to predict the time slot based on Day , Month , Sport chosen , it is also used to predict the sport based on timeslot , Day , Month . It is trained using the one hot encoding of the features since all the features are categorical. The model creates decision boundaries on the data points to classify. The training accuracy of the model is 0.29%. Figure 15 shows the decision tree till depth of 4.Feature Importance is found for features used to train the decision tree .

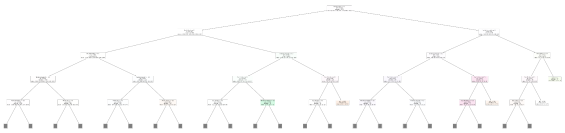


Figure 15. Decision Tree Classifier for sport prediction

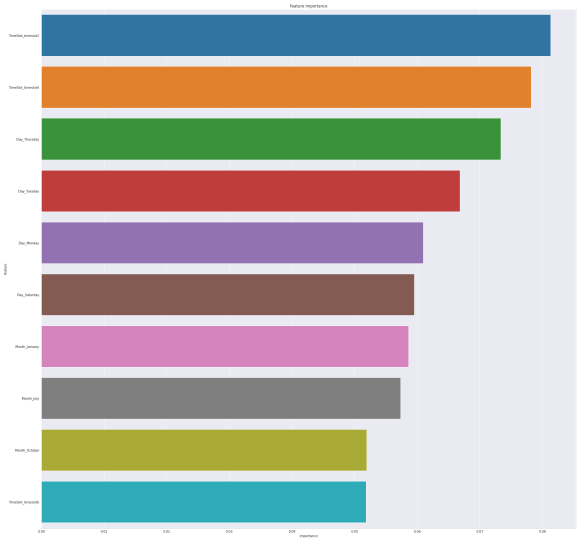


Figure 16. Feature Importance Sport prediction tree

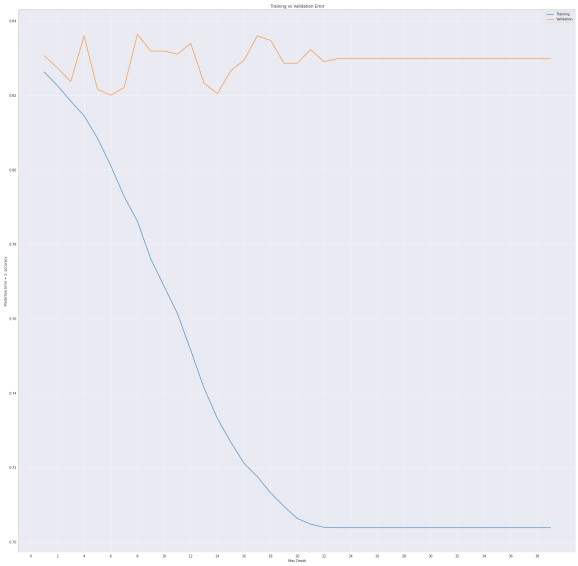


Figure 17. Tuning maximum depth of the tree for sport prediction.

timeslot :

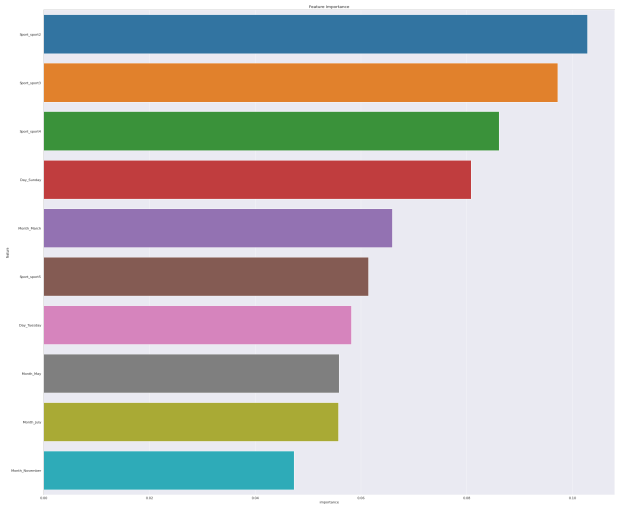


Figure 18. Feature importance for timeslot prediction.

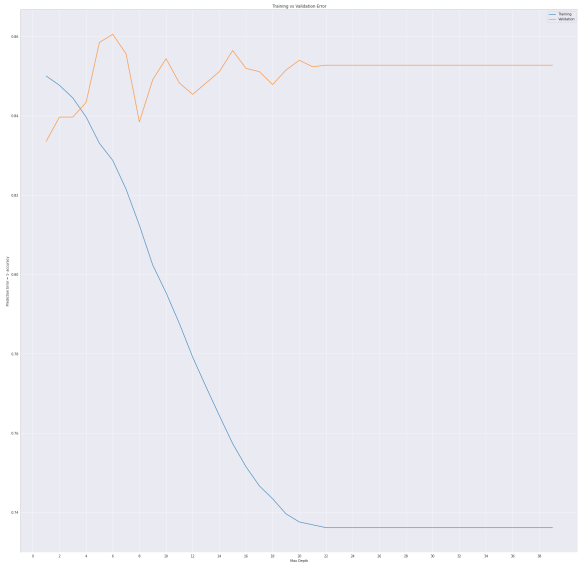


Figure 19. Tuning maximum depth of the tree for timeslot prediction.

Random Forest

feature importance for sports random forest model

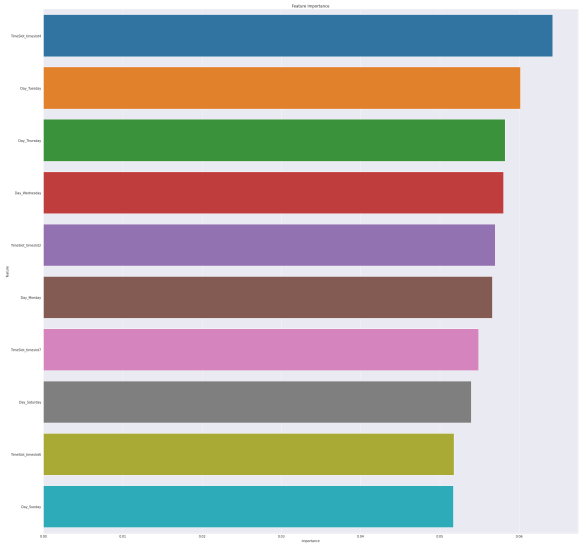


Figure 20. Feature Importance given by Random Forest model for sport prediction.

hyperparameter tuning for sports prediction model

```
rf_grid.best_params_
```

```
{'max_depth': 2, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimator
s': 100}
```

```
rf_grid.best_score_
```

```
0.17410049215738574
```

Figure 21. Hyperparameter tuning for sports prediction.

feature importance for sports random forest model

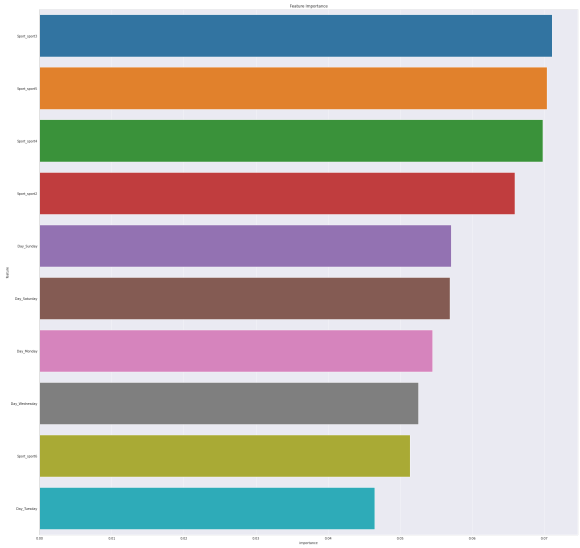


Figure 22. Feature Importance given by Random Forest model for time slot prediction.

hyperparameter tuning for timeslot prediction model

```
rf_grid.best_params_
```

```
{'max_depth': 8, 'max_features': 0.6, 'max_samples': 0.5, 'n_estimator
s': 60}
```

```
rf_grid.best_score_
```

```
0.1482200743727739
```

Figure 23. Hyperparemeter tuning for time slot prediction.

Voting Classifier

Models used in voting classifier include Logistic regression , random forest ,K Neighbors and Decision Tree Classifier

sports :

```
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
```

timeslot :

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

Figure 24. Accuracy of Voting Classifier.

Bagging Classifier

Decision Tree and Random forest models are used as base estimators in the bagging classifier . 500 base estimators are used , max sampling done is 25% and generating random subsets of training data is used. Model predicts sports with a 17.1% accuracy when random forest is the base estimator and with an accuracy of 16.8% when the decision tree is chosen as the base estimator . Model predicts the Time slot with a 15.1% accuracy with random forest as the base estimator and an accuracy of 14.7% when the decision tree is the base estimator.

Stacking

Stacking is an ensemble learning technique which combines the results of multiple models and feeds it to a meta learner that assigns weights to the base models to determine the better model for a particular set of data points. Random Forest classifier and the decision tree classifier are taken as base forest and the stacked model provided an accuracy of 17.65% for predicting sport and 14.4% for predicting timeslot.

sport :

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

Figure 25. Prediction of sport using Stacking model.

timeslot :

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

Figure 26. Prediction of timeslot using Stacking model.

ADA Boosting

ADA Boosting is an archetype of ensemble machine learning models where the stagewise additive method is used .Upsampling is used to increase the focus on incorrectly predicted data points. The weights are assigned to the models according to the mistakes made by it. It gives an accuracy of 15.6% while predicting the sport and after hyperparameter tuning, the best accuracy of 16.9% is given by the model when learning rate is set to 0.001 and 50 base estimators are used. The model gives an accuracy of 14.5% while predicting the timeslot, on hyperparameter tuning , the best results of 16.99% are given when 50 estimators with a learning rate of 0.001 are used.

Predicting the sentiment of user reviews for grounds using text classification.

Preprocessing of Dataset for Naive Bayes and SVM Models

The dataset consists of movie reviews from imdb in the form of text sentences. To make the data machine understandable , it has to be converted into numeric values. Before converting the data into numeric values, it can be preprocessed and filtered to remove unimportant words that do not add to the meaning of the final sentiment of the review. The words are converted lowercase to avoid creating two entries of the same word, short words like shouldn't have been elongated to should not have. The urls are removed from the sentences, special characters and multiple spaces are removed.HTML tags are removed. Stopword removal is a method of removing the most common English language words from sentences. These words appear quite frequently and do not add meaning to the sentiment of the sentence. A Bag Of Words Representation of words is used to represent the words in the form of vectors as explained in methodology.

word cloud for words adding to positive sentiment

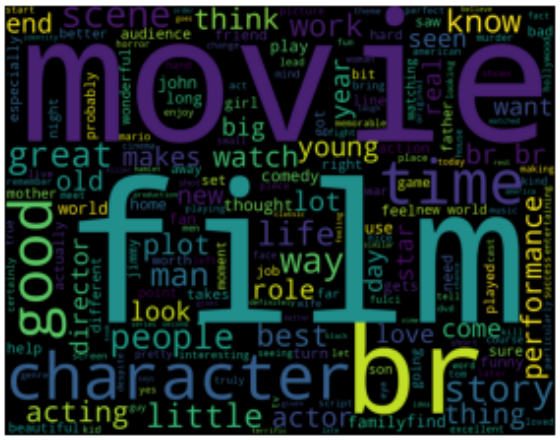


Figure 27. Word cloud representation of positive words.

word cloud for words adding to negative sentiment

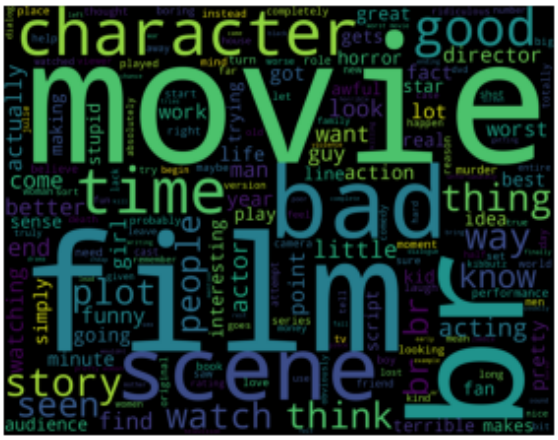


Figure 28. Word cloud representation of negative words.

Naive Bayes

```
NB :accuracy :  82.0


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

Figure 29. Naive Bayes sentiment classification model.

Support Vector Machine

```
SVM :accuracy :  82.75


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

Figure 30. SVM sentiment classification model.

Stacking : Stacking Naive Bayes Model and Support Vector Machine Model

Stacking is an ensemble learning technique in which multiple models are combined and the results are fed into a meta learner which assigns weights to models according to their performance .

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

Figure 31. Stacking model using NB and SVM.

Bag Of Words Representation and Feed Forward Text Classifier

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.

The sentence of the customer review will be represented as a vector with the vector element indicating the number of times that word occurs in the sentence. The vector length of all the reviews is kept the same by constructing a vocabulary of unique words occurring in all the reviews. These review vectors are inputted into the feed forward neural network where the first layer of the neural network will be of the size of the vocabulary. The final layer of the neural network will be of size 1 to determine 0 or 1 , where 0 indicates that the review was negative whereas the 1 indicates that the review was positive.

Neural Network training and testing ideologies include layers, activation , output , loss, gradient and learning update. The neurons are divided into several layers to train the model, the first layer is the inputs layer, then the hidden layer and the output layer. Here the input layer is of the size of the vocabulary of the bag of words representation of all reviews. Activation function is used at each layer to push the neuron to an active or inactive state based on the inputs and weights and biases from the previous connected neurons. ReLU (Rectified Linear Unit) and Sigmoid activation functions are used. The output layer has a single neuron whose activation value determines if the review is positive or negative. Forward propagation takes place on the input data and binary cross entropy is used to randomize and create subsets of data to be fed into the network , the weights and biases of the network is corrected using the gradient descent method where the local minima of the gradient of the loss function is calculated to minimize the loss. Adam optimizer is used to adjust the learning rate of the model according to the results provided by it. The model is trained over 10 epochs signifying that the training data is fed into the model for 10 times. Each time the loss is calculated and it is seen that the loss is reducing over the number of epochs. The train loss is the average of errors

made by the model neurons while predicting the sentiment.

```
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
```

Figure 32. Feed Forward Neural Network to predict sentiment .

Word Embeddings and RNN,LSTM,GRU

Word embedding representation is a better representation of text data as it shows the relation between the words more effectively. The Recurrent Neural Networks are used since the Feed Forward neural network does not read the data sequentially , in sentences sequence matters the most therefore RNN are used. But RNN has a problem of vanishing gradient, it does not remember the context of words at the beginning of the sentence. A solution to that was LSTM (Long Short Term Memory) , the input data from the previous hidden state is processed using gates and activation functions and stored in a memory channel maintained by the network and passed over, this helped recover from the vanishing gradient problem as the network remembers the earlier words, to make it more efficient GRU(Gated Recurrent Units) were used which have only two types of gates including forget and update. This also keeps track of the previous context but the two gates makes it more efficient to filter the inputs and perform operations on them.

Explain:

```
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
```

Figure 33. Gated Recurrent Unit Neural Network to predict Sentiment.

Voting, Bagging, Boosting

Voting of deep learning neural networks can be done by combining results of multiple neural networks and picking the most classified answer, this comes under Hard Voting. Soft voting can

also be used where the probability of the model predicting the output is taken into consideration . Bagging method is when multiple instances of the same neural network model is trained using different sets of input data and the result is aggregated using voting.
Boosting is a method where there can be multiple neural network models trained and the output of the previous model can be fed into the next network to learn about the mistakes made by the earlier model and recursively repeat the process.

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)

   trainable parameters include: Wh, bh

6. Conclusion

Ensemble learning techniques are useful when there are multiple models for a particular use case and each model finds an apt pattern for itself and provides good results for only a particular type of data points.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.

7. References

[1]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

[2]Heart Disease Prediction using Hybrid machine Learning Model, 2021
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.

[3]Music Genre Classification of audio signals Using Particle Swarm Optimization and Stacking Ensemble
K. Leartpantulak and Y. Kitjaidure, *"Music Genre Classification of audio signals Using Particle Swarm Optimization and Stacking Ensemble,"* 2019 7th International Electrical Engineering Congress (iEECON), 2019, pp. 1-4, doi: 10.1109/iEECON45304.2019.8938995.

[4]Time series regression and prediction based on boosting regression
Wen Gu, Baifeng Li, Baolong Niu, Wei Wei and Zhiming Zheng, *"Time series regression and prediction based on boosting regression,"* 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), 2014, pp. 251-254, doi: 10.1109/WARTIA.2014.6976244

[5]Multi-Class Text Classification: Model Comparison and Selection
W. Arshad, M. Ali, M. Mumtaz Ali, A. Javed and S. Hussain, *"Multi-Class Text Classification: Model Comparison and Selection,"* 2021 International Conference on Electrical,

Communication, and Computer Engineering (ICECCE), 2021, pp. 1-5, doi: 10.1109/ICECCE52056.2021.9514108.

[6]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

[7]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

[8]S. B. Rathod and T. M. Pattewar, "Content based spam detection in email using Bayesian classifier," 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 1257-1261, doi: 10.1109/ICCSP.2015.7322709.https://ieeexplore.ieee.org/document/7322709

[9] 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

[10]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[11]A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[12] V. Aakash;S. Sridevi;G. Ananthi;S. Rajaram; (2021). *Forecasting of Novel Corona Virus Disease (Covid‐19) Using LSTM and XG Boosting Algorithms . Data Analytics in Bioinformatics, (), –.* doi:10.1002/9781119785620.ch12 https://onlinelibrary.wiley.com/doi/10.1002/9781119785620.ch12

[13] Chen, Tianqi; Guestrin, Carlos (2016). *[ACM Press the 22nd ACM SIGKDD International Conference - San Francisco, California, USA (2016.08.13-2016.08.17)] Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16 - XGBoost. , (), 785–794.* doi:10.1145/2939672.2939785 https://dl.acm.org/doi/10.1145/2939672.2939785

[14] Swapan Talukdar, Kutub Uddin Eibek, Shumona Akhter, Sk Ziaul, Abu Reza Md. Towfiqul Islam, Javed Mallick, Modeling fragmentation probability of land-use and land-cover using the bagging, random forest and random subspace in the Teesta River Basin, Bangladesh,Ecological Indicators, Volume 126, 2021, 107612, ISSN 1470-160X, https://doi.org/10.1016/j.ecolind.2021.107612. https://www.sciencedirect.com/science/article/pii/S1470160X21002776

[15] Wei Chen, Haoyuan Hong, Shaojun Li, Himan Shahabi, Yi Wang, Xiaojing Wang, Baharin Bin Ahmad, Flood susceptibility modelling using novel hybrid approach of reduced-error pruning trees with bagging and random subspace ensembles Journal of Hydrology, Volume 575, 2019, Pages 864-873, ISSN 0022-1694, https://doi.org/10.1016/j.jhydrol.2019.05.089. https://www.sciencedirect.com/science/article/pii/S0022169419305347

[16] Santosh Singh Rathore and Sandeep Kumar. 2016. A Decision Tree Regression based Approach for the Number of Software Faults Prediction. SIGSOFT Softw. Eng. Notes 41, 1 (January 2016), 1–6. DOI:https://doi.org/10.1145/2853073.2853083 https://dl.acm.org/doi/abs/10.1145/2853073.2853083

[17] Min Xu, Pakorn Watanachaturaporn, Pramod K. Varshney, Manoj K. Arora, Decision tree regression for soft classification of remote sensing data, Remote Sensing of Environment, Volume 97, Issue 3, 2005, Pages 322-336, ISSN 0034-4257, https://doi.org/10.1016/j.rse.2005.05.008. https://www.sciencedirect.com/science/article/pii/S0034425705001604

[18] Robert Tibshirani,Regression Shrinkage and Selection Via the Lasso,Journal of the Royal Statistical Society: Series B (Methodological)Volume 58, Issue 1 p. 267-288,First published: 1996,https://doi.org/10.1111/j.2517-6161.1996.tb02080.x https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x

[19] Ciuperca, G. Model selection by LASSO methods in a change-point model. *Stat Papers* 55, 349–374 (2014). https://doi.org/10.1007/s00362-012-0482-x https://link.springer.com/article/10.1007/s00362-012-0482-x#citeas

[20] Norman R. Draper & R. Craig Van Nostrand (1979) Ridge Regression and James-Stein Estimation: Review and Comments, Technometrics, 21:4, 451-466, DOI: 10.1080/00401706.1979.10489815 https://www.tandfonline.com/doi/citedby/10.1080/00401706.1979.10489815?scroll=top&needAccess=true

[21] Hsu, D., Kakade, S.M. & Zhang, T.. (2012). Random Design Analysis of Ridge Regression. Proceedings of the 25th Annual Conference on Learning Theory in Proceedings of Machine Learning Research http://proceedings.mlr.press/v23/hsu12.html

[22] Dr. Swagato Chatterjee, Explaining customer ratings and recommendations by combining qualitative and quantitative user generated contents, Decision Support Systems, Volume 119, 2019, Pages 14-22, ISSN 0167-9236, https://doi.org/10.1016/j.dss.2019.02.008. https://www.sciencedirect.com/science/article/pii/S0167923619300363

[23] Geetha, M.; Singha, Pratap; Sinha, Sumedha (2017). *Relationship between customer sentiment and online customer ratings for hotels - An empirical analysis. Tourism Management, 61(), 43–54.* doi:10.1016/j.tourman.2016.12.022 https://www.sciencedirect.com/science/article/abs/pii/S0261517716302709

[24]X. Jin, X. Hou and C. Liu, "Multi-class AdaBoost with Hypothesis Margin," 2010 20th International Conference on Pattern Recognition, 2010, pp. 65-68, doi: 10.1109/ICPR.2010.25.