HIGHLIGHTS IN CLASSIFICATION TECHNIQUES

Shaimaa Mohamed, Aya Ahmed, Rana Ahmed
Dept. Computer Engineering
University of Ain Shams,19P7484@eng.asu.edu.eg,
19P1689@eng.asu.edu.eg,19P2468@eng.asu.edu.eg

ABSTRACT

Objective our goal is build model that determine the country destination of user by applying data mining technique to features selected from data. That help us to guarantee the best results
Background Instead of waking to overlooked 'Do not disturb' signs, Airbnb travelers find themselves rising with the birds in a whimsical tree house, having their morning coffee on the deck of a houseboat, or cooking a shared regional breakfast with their hosts.
New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. By accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, decrease the average time to first booking, and better forecast demand.
Methods: we apply 4 different technique which are KNN, decision tree, Naive Bayes, and random forest to help us to compare between the results and choose the best technique to apply.
Results: we see that the best classifier is Naive Bayes that gives the highest accuracy after that KNN then random forest and the last one is the decision tree.
Conclusion: to sum up the best classification technique to apply for our data set is Naive Bayes as when we train the model using test data gives the highest accuracy.

INTRODUCTION

Kaggle offers a really competitive and amazing challenges like Airbnb challenge that we intend to solve using data Mining Techniques. Our solutions will mainly focus on predicting the country destination for the new passengers.

The dataset is very large and challenging to mine, so we will be accurate in making data preprocessing and choosing the appropriate techniques to get high accuracy for the model.

Our steps: 1) Finding a model that explains and differentiates various data classes and concepts is the task of classification, which falls under the category of data analysis. On the basis of a training set of data that includes observations and whose category membership is known, classification is the problem Of determining which of a set of categories (subpopulations), a new observation belongs to.
2. By allowing the model to learn using the provided training set, several algorithms are employed to construct a classifier. For reliable outcome prediction, the model must be trained. Steps in classification include using a model to predict class labels, testing the model using test data, and determining the degree to which the classification rules are accurate.

The purpose of this research paper is making experiments, trying to get best results, changing in hyper parameters, and decreasing the computations.

DATASET DESCRIPTION

We are given a list of users along with their demographics, web session records, and some summary statistics, training and test sets are split by dates.
Files in dataset: **countries.csv** - summary statistics of destination countries in this dataset and their locations.
age_gender_bkts.csv - summary statistics of users' age group, gender, country of destination,

sample_submission.csv - correct format for submitting your predictions,
sessions.csv - web sessions log for users,
train_users.csv - the training set of users

Note: There are 12 possible outcomes of the destination country: 'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL','DE', 'AU', 'NDF' (no destination found), and 'other'. Here 'NDF' is different from 'other' as 'other' means there was a booking, but is to a country not included in the list, while 'NDF' means there wasn't a booking.

KAGGLE PROBLEM LINK

Airbnb New User Bookings | Kaggle

EVALUATION CRITERIA

Make Testing by splitting our train_users file into 70% training , 30 % testing and start measuring the accuracy using metrics for our resulted model, and is the model fitting or overfitting data.
The Evaluation criteria will be based on selecting the minimum features that will give us higher accuracy, trying different transformations or data reduction in order to manipulate the data and get efficient results, computation of the algorithm shall be fast and reach high precision.

APPROACH

Our aim is to predict which country destination will be selected in user's first booking, and see frequent item sets using the train data file that we have.

Preprocessing

Preprocessing data removes missing or inconsistent data values resulting from human or computer error, which can improve the accuracy and quality of a dataset, making it more reliable.
It makes data consistent. When collecting data, it's possible to have data duplicates, and discarding them during preprocessing can ensure the data values for analysis are consistent, which helps produce accurate results.
It increases the data's algorithm readability. Preprocessing enhances the data's quality and makes it easier for machine learning algorithms to read, use, and interpret it.

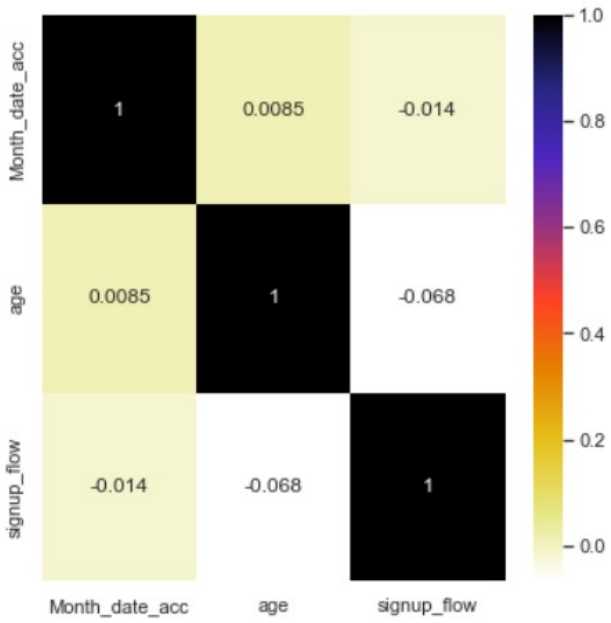
Data Cleaning

This is the first step which is implemented in Preprocessing. In this step, the primary focus is on handling missing data, noisy data, detection, and removal of outliers minimizing duplication, and computed biases within the data.

Approach

- First we found that the train_data requires preprocessing by getting its information, and description by python simple functions.
- We checked on the null values by using isnull ().sum function, and we found that the age, data_of_first_booking and first_Affiliate_tracked contain many null values.
- As the age is from our data we are interested in in signing up for booking , so we filled the missing values by the median and excluded ages above 95 and less than 15 , visualization by distplot was a good choice to see age data after cleaning and removing outliers
- Then we removed all the null values, and started checking on having duplicates and also removing them.
- We changed all our dates to be in date-time format , and see if we could drop any un useful columns and continue doing visualizations trying to figure out relations between country_destination what we want to predict and the age or the count of the passengers going to a certain destination , see pie chart of their signing up methods and apps used.
- We tried getting the correlation between the column attributes to help us in eliminating some features, we found a correlation between time_of_first_stamp and date of first booking, the date of first_booking and date of account created, moreover between timestamp and data_of_account created.

- Correlation interpretation



Missing values

Here are a few ways to solve this issue:

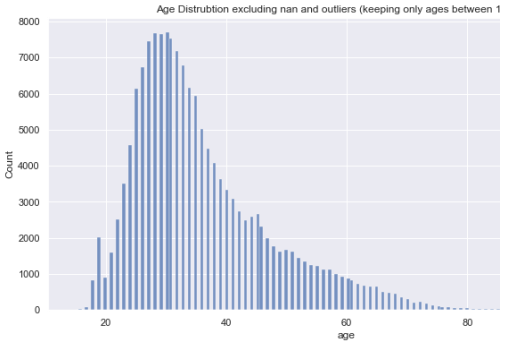
Ignore those tuples

This method should be considered when the dataset is huge and numerous missing values are present within a tuple.

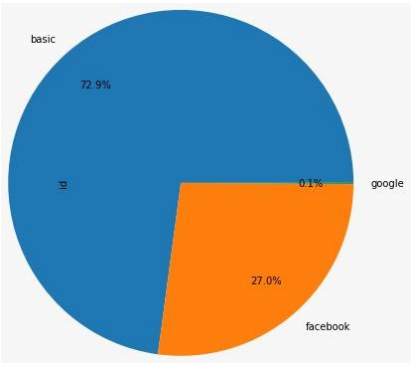
Fill in the missing values

There are many methods to achieve this, such as filling in the values manually, predicting the missing values using regression method, or numerical methods like attribute mean.

VISUALIZATION AFTER CLEANING

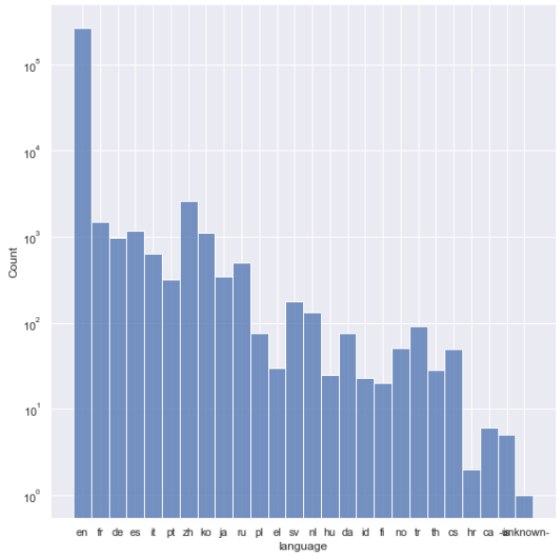


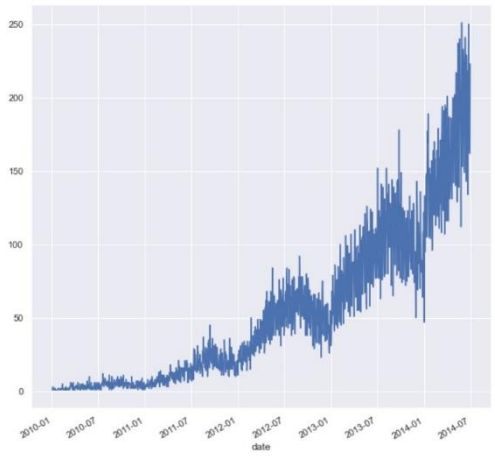
Visualization for the age values after filling the missing values with the median and discarding the ages below 15 and above 95 years.



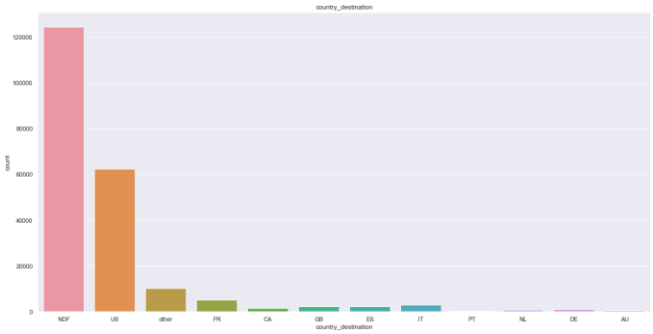
This graph shows the sigup_method results after cleaning the data , the majority uses basic , and the lowest percentages goes to google in signing up.

This visualization shows the most sign up language was done in English





This graph illustrates the number of passengers, according to the years of creating account, and this helps us to know which years reach the peak in case of booking.



This graph illustrates that majority of passengers, their destination is not found, then the US was the highest destination.

This graph shows the relation between the language and the destination country, so we conclude that US, other are the most countries that have passenger signing up in many languages.

DATA MINING TECHNIQUES THAT WE USE:

As our data is labeled so, we will use supervised learning techniques like Classification Techniques.

Decision Trees: It is one way to display an algorithm that only contains conditional control statements till reaching the goal so, we can ask many questions until reaching the right classification for our problem (predicting the destination country). The reasons why we chose the decision tree are that it requires the least data preparation, the cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree, and it's capable of performing multi-class classification on a dataset. [5]

Also it is used to handle multi-output problems. Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by Boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret. Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated [5].

Random Forest: A classifier that uses multiple decision trees on different subsets of a given dataset and averages the results to increase the predicted accuracy of that dataset. The random forest gathers the results from each decision tree and bases its expectation of the final result on the majority votes of the predictions, as opposed to relying solely on one decision tree.

In random forests, hyper parameters are either used to speed up the model or to improve its performance and predictive ability. The predictive power is increased by using the following hyper parameters: n estimators: The number of trees the algorithm constructed prior to averaging the results. Max features: Maximum features that a random forest can use before splitting a node.

Minimal number of leaves necessary to split an internal node is determined by the mini sample leaf function.

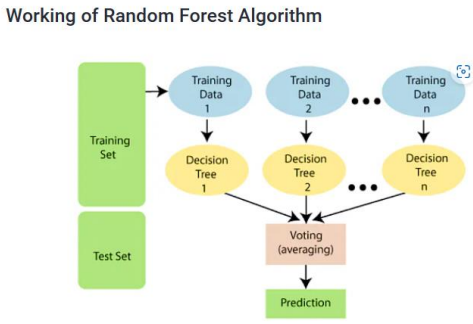


IMAGE COURTESY: javapoint

The following steps explain the working Random Forest Algorithm:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

We tried this algorithm to see the contrast between it and the decision tree, see if it leads to a great increase in accuracy or not. [6]

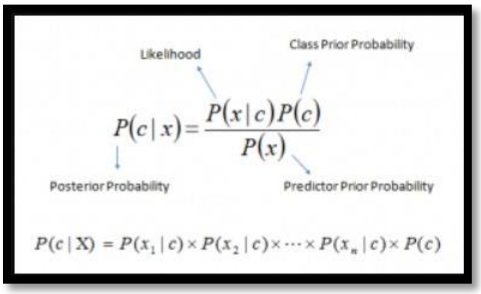
Comparison

Decision Trees	Random Forest
• They usually suffer from the problem of overfitting if it's allowed to grow without any control.	• Since they are created from subsets of data and the final output is based on average or majority ranking, the problem of overfitting doesn't happen here.
• A single decision tree is comparatively faster in computation.	• It is slower.
• They use a particular set of rules when a data set with features are taken as input.	• Random Forest randomly selects observations, builds a decision tree and then the result is obtained based on majority voting. No formulas are required here.

KNN:
K-nearest Neighbors Classifier, We select the k entries in our training data set which are closest to the new sample then we find the most common classification of these entries to be our final predict. Reasons for choosing it -> using the K-NN algorithm, new data can be quickly and accurately classified into a suitable category. Since K-NN is a non-parametric algorithm, it makes no assumptions about the underlying data. As we have a large amount of data we specified n_neighbors to 21. [4]

Naive-Bayes:

The naive Bayes Algorithm helps to categorize the data based on the computation of conditional probability values. It uses class levels represented as feature values or vectors of predictors for classification. It is a quick algorithm for classification issues and is simple to use and more scalable in case of large datasets like our train data. To calculate the posterior probability P(c|x), it is helpful to use the prior probabilities of the class P(c), the predictor P(x), and the probability of predictor given class, also called as likelihood P(x|c). [3]



RESULTS AND ANALYSIS

By splitting the train data into 70 % training and 30 % testing we get

1) ***Appling decision tree***

We made the **decision tree** criterion which is the entropy and fitted the model. The decision tree takes as input two arrays: an array X, sparse or dense, of shape (n_samples, n_features) holding the training data, and an array Y of integer values, shape (n_samples), holding the class labels for the training samples. [1]
As we applied the decision tree first, we categorized all data and transformed the dates by timestamp functions and we have chosen features like age, gender and date_of_account created. We found that the accuracy was 0.5403235057932775. Our Analysis here is that these features didn't achieve accurate results as we have some other features we exclude them by trial, error and by seeing a correlation.

2) ***Applying KNN***

Appling KNN approach -> **we** made first the neighbors by 3 and started fitting the model and predicting on test data, accuracy was 0.6983289357959542.
Analysis of KNN By increasing the n-neighbors to 17, the accuracy increased to be 0.7526289625635731.

3) ***Naïve Bayes***

Applying Naïve Bayes in order to increase the accuracy of the model and make correct predicts, we found that with the same 3 features we tried before we get
0.7536231884057971 accuracy. Our Analysis we found that to increase this accuracy is to standardize the data so, we should use transformation or different methods to convert it in normal distribution. Also, we noticed that Naive Bayes classifiers have few options for parameter tuning, such as smoothing with alpha=1, learning class prior probabilities with fit prior=[True|False], and other options.

4) ***Random Forest***

Applying Random Forest algorithm, the accuracy was 0.5891935298841344 when putting the estimators by 100, and when we increase the estimator by 200 we get accuracy 0.5893847271614853. Our Analysis we found that first, random subsets of training samples are used in each tree training in the sample. Second, the randomly selected features of the unpruned tree nodes are used to select the best split, so we need to be more determinant in choosing features and limit branching for saving computations.

DEVELOPMENT

1st improvement: From the shortcomings that was stated before is the problem of feature selection and was the three features chosen before enough or not?, the solution for this problem is that we applied the **LDA algorithm** which is a method in feature reduction, a classifier created by fitting class conditional densities to the data and applying Bayes' rule, with a linear decision boundary. The model assumes that all classes have the same covariance matrix and fit a Gaussian density to each class. [2]

It uses the mean values of the classes and maximizes the distance between them. It uses variation minimization in both the classes for separation. By projecting the input to the most discriminative directions, the fitted model can also be used to reduce the dimensionality of the input.

So, we started putting the number of features we want and apply LDA with the number of components that we want to enhance our results and accuracy.

By increasing the components, the accuracy increases so, we have chosen the number of components is 3.

Three key benefits of performing feature selection on your data are:

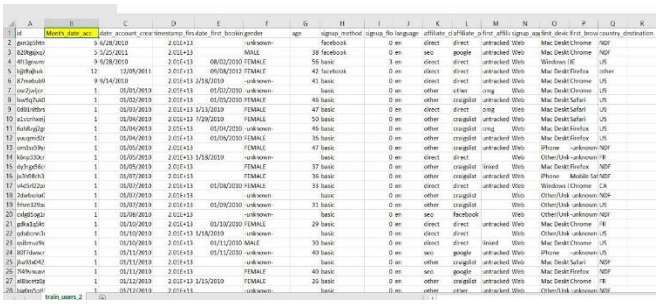
Reduces Over fitting: Less redundant data means less opportunity to make decisions based on noise.

Improves Accuracy: Less misleading data means modeling accuracy improves.

Reduces Training Time: Less data means that algorithms train faster.

2nd improvement: Our data contains columns with date formats and actually the date is long, as it consists of day, month, and year. So it will take a long time in making decision tree based on this three sections in order to classify by date, so tree gets more complex , and takes much time for computation.

The solution is that we have a new column in the train_users excel file to extract the month of the date_of_account_created , and by this way we can put the month in our features instead of the whole date , and this really enhanced the accuracy in each algorithm.



This is the image of the excel sheet after adding the Month_date_acc column to enhance the results and make optimization in the decision tree.

3rd improvement:

A crucial step in modelling the algorithms with the datasets is scaling the features. The data that is typically used for modelling purposes is obtained using a variety of methods such as Questionnaire, Surveys. So, by formatting the statistical distribution of the data as follows, standardization, a scaling technique, renders the data scale-free by making mean =0 and standard deviation = 1.

RESULTS AFTER IMPROVEMENTS

-Choosing more features like month, age, gender, signup method, signup flow then apply LDA [2].

-Decision tree precision increased to 0.6652900462697411.

-KNN classifier precision become 0.6992084432717678 with k-neighbors by 19 and when k=21 , the accuracy 0.6993.

-Naïve Bayes increased to a very good accuracy 0.9991204925241864 which it represents the perfect algorithm in this problem.

-Random Forest Algorithm increased to 0.6827272379641314 which is greater than decision tree as we proved that the Random Forest algorithm is more optimized that decision trees.

RESULTS AFTER ADDING LANGUAGE FEATURE

We thought that language can be a feature that can increase our accuracy because it wasn't correlated with any other

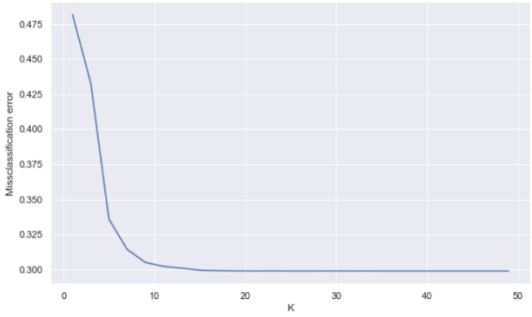
features, and by graphing it with the destination, we got some interpretation, so we concluded that it can influence our decision in booking our country destination.

But the results weren't so significant than before as it doesn't affect the accuracies value as we expect

1. Random forest becomes 0.681885969943788
2. Decision Tree becomes 0.6613513823563153
3. KNN becomes 0.6992849221827081
4. Naive Bayes becomes 0.9994646476234178

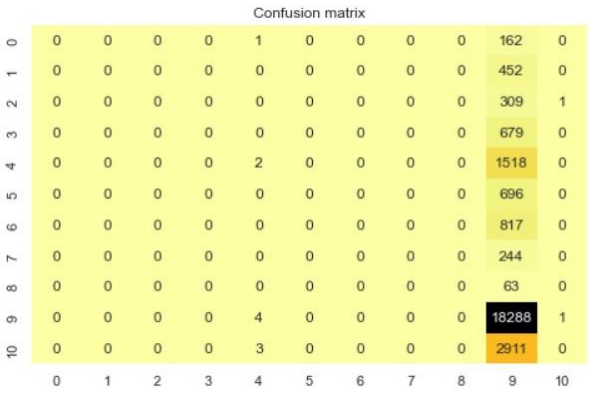
Interpretation for KNN

We found all the accuracies increased except by applying KNN after the improvements decreased from its accuracy, this is due to not choosing the right k, and we have known this results by applying the MSE techniques and get this graph between the misclassification error and the value of k.



NAÏVE BAYES RESULTS

This the covariance matrix that shows the true positive, true negatives and false positive, false negatives.

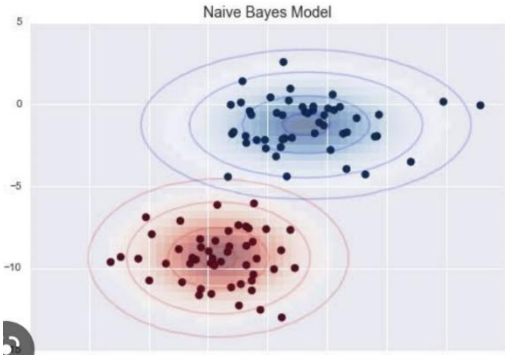


	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	1	0	0	0	0	162	0
1	0	0	0	0	0	0	0	0	0	452	0
2	0	0	0	0	0	0	0	0	0	309	1
3	0	0	0	0	0	0	0	0	0	679	0
4	0	0	0	0	2	0	0	0	0	1518	0
5	0	0	0	0	0	0	0	0	0	696	0
6	0	0	0	0	0	0	0	0	0	817	0
7	0	0	0	0	0	0	0	0	0	244	0
8	0	0	0	0	0	0	0	0	0	63	0
9	0	0	0	0	4	0	0	0	0	18288	1
10	0	0	0	0	3	0	0	0	0	2911	0

CONCLUSION

Going throughout this project, we learnt how to deal with the data preprocessing, data transformation, how to make feature reduction, Standardization, and finally using different techniques of classification, in order to make a model, by splitting the data into x_train that represents all the features except the destination and y_train will be the country destination in order to make proper prediction and the accuracy of the model is the measure of how my model gets the results correct.

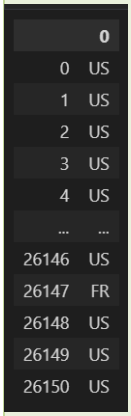
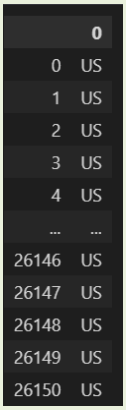
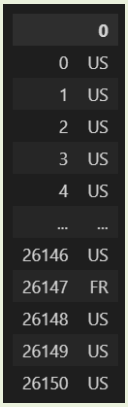
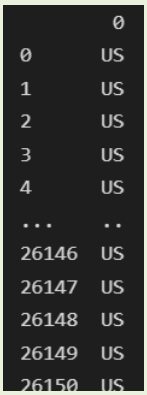
To sum up, first we apply decision tree but we categorize the data and add new columns for data set that extract only the month from the data to decrease probability of date and increase accuracy of the technique, then we choose feature as age, gender, month_date_account, and signup method, but we apply LDA to decrease the dimensionality of features. After that apply the technique on 70% of data and test the reminding and this give accuracy 0.6652 Second we apply KNN algorithm but with using of different n-neighbors till reach the highest accuracy with n-neighbors value 17 which is 0.7526. Third, the best algorithm used so far is naive Bayes using the same 3 features using before from LDA and give accuracy 0.999.



Fourth we used random forest, as it enhanced in the decision tree results by getting the optimal of applying more than one decision trees and the accuracy result was 0.684.the best algorithm used here was naïve Bayes which is based on conditional probability of values.

The second algorithm was KNN, then random forest and finally the decision tree.

SUMMARY

Points	Decision Tree	KNN	Naïve Bayes	Random Forest
Algorithm	We can ask many questions until reaching the right Classification for our problem.	We select the k entries which are closest to the new sample then we find the most common classification to be our final predict.	It uses class levels represented as feature values or vectors of predictors for classification.	A classifier that uses multiple decision trees on different subsets of a given dataset and averages the results to increase the predicted accuracy of that dataset.
Accuracy	0.665290046269971	0.6992084432717678	0.9991204925241864	0.6827272379641314
Predicate				

GITHUB LINK

https://github.com/Shaimaa-moh/Mining-Project

REFERENCES

[1] Navlani, A. (2018, December 28). Python decision tree classification tutorial: Scikit-Learn Decisiontreeclassifier. DataCamp. Retrieved January 4, 2023, from https://www.datacamp.com/tutorial/decision-tree-classification-python.

[2] Learn. scikit. (n.d.). Retrieved January 4, 2023, from https://scikit-learn.org/0.16/

[3] Nayak, V. (2021) Author identification with naive Bayes algorithm, Medium. Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/author-identification-with-naive-bayes-algorithm-2-8b43854c1429 (Accessed: January 4, 2023).

[4] K-Nearest Neighbor (KNN) algorithm for Machine Learning - Javatpoint. www.javatpoint.com. (n.d.). Retrieved January 4, 2023, from https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

[5] Understanding the decision tree structure. scikit. (n.d.). Retrieved January 4, 2023, from https://scikit-learn.org/stable/auto_examples/tree/plot_unveil_tree_structure.html#sphx-glr-auto-examples-tree-plot-unveil-tree-structure-py.

[6] Sklearn.ensemble.randomforestclassifier. scikit. (n.d.). Retrieved January 4, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

.