DEPARTMENT OF STATISTICS

CERTIFICATE

Certified that this is the bonafide record of project work done by Mr. Shrajan Kumar during the year 2023 as a part of his M.Sc (Statistics) fourth semester course work.

Reg. No.	**2**	**1**	**4**	**0**	**1**	**4**

Project Guide **Head of the Department**

Examiner

1.

2.

Place: Ujire
Date:

Predictive Analysis on Attempts Range to Solve Problems for Enhancing User Engagement

Project Report submitted to the

SDM Post Graduate Centre, Ujire

in partial fulfilment of the degree of

MASTER OF SCIENCE
IN
STATISTICS

by

Shrajan Kumar

Under the supervision of

Ms. Supriya S.P.

Assistant Professor
Department of Post Graduate Studies in Statistics

SRI DHARMASTHALA MANJUNATHESHWARA

COLLEGE (Autonomous)

UJIRE - 574240

Karnataka, INDIA

July 2023

DECLARATION

I, Shrajan Kumar, hereby declare that the matter embodied in this report entitled '**Predictive Analysis on Attempts Range to Solve Problems for Enhancing User Engagement**' is a bonafide record of project work carried out by me under the guidance and supervision of **Asst. Prof. Supriya S.P**, Department of Statistics, SDM College, Ujire - 574240, Karnataka, India. I further declare that no part of the work contained in the report has previously been formed the basis for the award of any Degree, Diploma, Associateship, Fellowship or any other similar title or recognition of any other university.

Date: (Shrajan Kumar)
Place: Ujire E-mail: shrajankumar45@gmail.com

CERTIFICATE

This is to certify that the project report entitled **Predictive Analysis on Attempts Range to Solve Problems for Enhancing User Engagement**' is a bonafide record of an authentic work carried out by **Shrajan Kumar**, under my guidance and supervision in the Department of Post Graduate Studies and Research in Statistics, SDM College, Ujire, in partial fulfilment of the requirements for the award of the degree of Master of Science in Statistics, under Mangalore University, Mangalagangothri. I further certify that this report or part thereof has not previously been presented or submitted elsewhere for the award of any Degree, Diploma, Associateship, Fellowhsip or any other similar title of any other institution or university.

Date: (Supriya S.P)
Place: Ujire E-mail: supriyasp@sdmcujire.in

ACKNOWLEDGEMENTS

Firstly, I would like to thank our Principal ***Dr. B.A. KumarHegde*** *for providing the necessary facilities for the completion of this project work in our college. I would also thank our Dean* ***Dr. P. Vishwanatha*** *for his support. It is my privilege to thank our HOD* ***Prof. Shanthiprakash*** *for his suggestions and support. I am very grateful to my Research Supervisor,* ***Asst. Prof. Supriya S.P.****, Department of pg studies in Statistics, SDM College, Ujire, for her kind help and encouragement throughout my project work.*

I gratefully acknowledge my teachers at the Department of Statistics, SDM College, Ujire, ***Asst. Prof. Pradeep K*** and ***Asst. Prof. Shwetha Kumari*** *for their support during my project work.*

I am also thankful to all my family members and friends for their constant encouragement and help in each step.

My sincere thanks also goes to the students of SDM College, Ujire, who have helped me directly or indirectly during my project work.

*Finally to all who helped me in many ways, I say, '****Thank You****!'.*

(Shrajan Kumar)

Contents

List of Tables

List of Figures

1 Chapter 1
INTRODUCTION

1.1 Introduction

Online judges provide a dynamic platform where users, ranging from beginners to experts in competitive programming, engage in problem-solving activities to enhance their programming skills. With a diverse user base, each individual possesses varying levels of proficiency in different problem categories such as graph algorithms, dynamic programming, and more. Identifying patterns within these user behaviours can unlock valuable insights and pave the way for a better understanding of their programming tendencies. This project aims to leverage statistical techniques and modern machine learning approaches to model user behaviour and predict the range of attempts a user will make when solving a given problem, based on user and problem details.

By discerning patterns from historical user data, it becomes possible to uncover underlying trends and tendencies. Such insights hold immense value for the programming committee, enabling them to suggest relevant problems tailored to individual users' skill sets and interests. Moreover, the identification of these patterns can empower the committee to automatically provide hints and guidance to users who may encounter challenges while attempting certain problems.

To achieve these goals, the project will employ a combination of statistical techniques and state_of_the_art machine learning methodologies. Statistical techniques will aid in the exploratory analysis of user and problem data, allowing for an initial understanding of the dataset's structure and characteristics. This analysis will involve identifying potential relationships, trends, and distributions within the data.

Additionally, modern machine learning approaches will be utilized to develop predictive models. These models will leverage the identified patterns to estimate the range of attempts a user is likely to make while solving a particular problem. The predictive models can be trained using supervised learning techniques, utilizing historical user data that includes details such as problem complexity, user experience level, problem category, and previous attempts made.

By deploying these predictive models within the online judge platform, the programming committee will be equipped with valuable tools to enhance user experience. Relevant problem suggestions based on each user's skill level and preferred problem categories can be provided, ensuring an engaging and tailored learning experience. Furthermore, automatic hints and guidance can be generated based on the predicted user behaviour, offering timely support and facilitating skill development.

Overall, this project endeavours to unlock the potential embedded within the vast amounts of user and problem data generated on online judge platforms. By employing statistical techniques and modern machine learning approaches, the project aims to model user behaviour, predict the range of attempts for problem solving, and provide valuable insights for the programming committee to enhance the learning journey of users. The expected outcomes of this project include improved user engagement, enhanced problem recommendations, and automatic assistance, all of which contribute to an enriched learning experience for users of the online judge platform.

1.2 Literature Review

In 2014 Bhumika Bhatt, Prof.Premal J Patel, Prof.Hetal Gaudani published the paper named A Review Paper on Machine Learning Based Recommendation Engine for customer recommendation system.In this paper authors represents the overview of Approaches and techniques generated in recommendation system. Recommendation system is categorized in three classes: Collaborative Filtering, Content based and hybrid based Approach. This paper classifies collaborative filtering in two types: Memory based and Model based Recommendation .The paper elaborates these approaches and their techniques with their limitations.

On September 2012 Jehad Ali1, Rehanullah Khan, Nasir Ahmad published a paper Random Forest and decision tree in which authors have compared the classification results of two models i.e, Random Forest and the J48 for classifying twenty versatile datasets.They took 20 data sets available from UCI repository containing instances varying from 148 to 20000.They compared the classification results obtained from methods i.e. Random Forest and Decision Tree(J48).The classification parameters consist of correctly classified instances,incorrectly classified instances, F-Measure, Precision, Accuracy and Recall.The classification results show that Random Forest gives better results for the same number of attributes and large data sets i.e. with greater number of instances, while J48 is handy with small data sets (less number of instances). The results from breast cancer data set depicts that when they increased the number of instances from 286 to 699,the percentage of correctly classified instances increased from 69.23 percentage to 96.13 percentage for Random Forest i.e.for dataset with same number of attributes but having more instances,the Random Forest accuracy increased.

In 2008 June Yunhong Zhou, Dennis Wilkinson, Robert Schreiber and Rong Pan carried out a research called LargeScale Parallel Collaborative Filtering for the Netflix. In this paper, they describe Alternating Least Squares with Weighted λ Regularization (ALSWR), a parallel algorithm that they designed for the Netflix Prize, a large-scale collaborative filtering challenge. They used parallel Matlab on a Linux cluster as the experimental platform. Their work showed empirically that the performance of ALSWR monotonically increases with both the number of features and the number of ALS iterations. They applied ALSWR to the Netflix dataset with 1000 hidden features obtained a RMSE score of 0.8985,which is one of the best results based on a pure method.Combined with the parallel version of other known methods,they achieved a performance improvement of 5.91

In 2019 November Candice Bente,JacaAnna Csorg carried out the research called Comparative Analysis of XGBoost in this study researchers present an empirical analysis of XGBoost, a method based on gradient boosting that has proven to be an efficient challenge solver. Specifically,the performance of XGBoost in terms of training speed and accuracy is compared with the performance of gradient boosting and random forest under a wide range of classification tasks. In addition, the parameter tuning process of XGBoost was thoroughly analyzed.The results of this study show that the most accurate classifier in terms of the number of problems with the best performance in the problems investigated,was gradient boosting.Nevertheless,the differences with respect to XGBoost and to random forest using the default parameters are not statistically significant in terms of average ranks.They observed that XGBoost and gradient boosting trained using the default parameters of the packages were the least successful methods. In consequence they concluded that a meticulous parameter search is necessary to create accurate models based on gradient boosting.This is not the case for random forest, whose generalization performance was slightly better on average when the default parameter values were used .In fact tuning in XGBoost the randomization parameters sub-sampling rate and the number of features selected at each split was found to be unnecessary as long as some randomization is used. In their experiments,researchers fixed the values of the subsampling rate to 0.75 without replacement,reducing the size of the parameter grid search 16fold and improving the average performance of XGBoost.Finally,from the experiments of this study, which are based on grid search parameter tuning using within train 10 fold cross-validation, the tuning phase contributed to over 99.9 percentage of the computational effort necessary to train gradient boosting or XGBoost.Finally they concluded that XGBoost allows for a fine parameter tuning using acomputationally efficient algorithm. This is not as feasible with random forest(as small gains are obtained, if at all, with parameter tuning) or with gradient boosting, which requires longer computational times.

In November 2013 Aida Ali1,Siti Mariyam Shamsuddin and Anca L has presented a review paper Classification with class imbalance problem in which they explained about classification technique for imbalanced data set.Most existing classification approaches assume the underlying training set is evenly distributed.In class imbalanced classification,the training set for one class (majority) far surpassed the training set of the other class (minority) in which,the minority class is often the more interesting class. In this paper,researcher reviewed the issues that come with learning from imbalanced class data sets and various problems in class imbalance classification.They also presented A survey on existing approaches for handling classification with imbalanced datasets.

1.3 Objectives

The objectives of this project are as below :

- To study whether attempts range is affected by number of problems solved.
- To study the association between user rank and attempts range.
- To study the influence of ratings on attempts range.
- To determine the influence of problem points on attempts range.
- To build a model which predicts the attempts taken by a user to solve a given problem, provided the user current status and problem details.

1.4 Scope of Study

Recommending the questions that a programmer should solve given his/her current expertise is a big challenge for Online Judge Platforms but it is a essential task to keep a programmer engaged on their platform.So it is very important to predict the number of attempt that a user takes to solve a given problem.Machine learning approach to this will solve the problem easily. Without much cost, energy and time, ML models can be used to predict the no of attempts taken to solve a problem given the user current status.

2 Chapter 2
METHODOLOGY

2.1 Materials and Methods

2.1.1 About the Data

A secondary data is collected from the Analytics Vidhya competition which contains three data sets named as train submissions,user data and problem data.Data set train submission contains 1,55,295 submissions detail of users.Data set named user data has 3571 rows and 11 columns containing data of users.Data set called problem data has 6544 rows and 4 columns,this file contains data of the problems. All data sets are in csv format.Description about the variable are as follows:

Dataset user_data:

- **user id:** Unique ID assigned to each user.
- **submission count:** Total number of user submissions.
- **problem solved:** Total number of accepted user submissions.
- **contribution:** User contribution to the judge.
- **country:** Location of user.
- **follower count:** Amount of users who have this user in followers.
- **last online time seconds:** Time when user was last seen online.
- **max rating:** Maximum rating of user.
- **rating:** Rating of user.
- **rank:** Can be one of ”beginner” ,”intermediate”,”advanced”,”expert”.
- **registration time seconds:** Time when user was registered.

Dataset problem_data:

- **problem id:** Unique ID assigned to each problem.
- **level id:** The difficulty level of the problem between ‘A’ to ‘N’.
- **points:** Amount of points for the problem.
- **tags:** Problem tags like greedy, graphs, DFS etc.

Dataset final_submission:

- **‘attempts_range’;** Denotes the range no.In which attempts the user made to get the solution accepted lies.

Below given criteria is used to define the attempts range.

Table 1: Table of encoded Attempt Ranges and their corresponding number of attempts

Attempts Range:	**No. of Attempts**
1	1–1
2	2–3
3	4–5
4	6–7
5	8–9
6	≥10

2.1.2 Statistical Techniques

- **Bag of Words**: Bag of words is the most trivial representation of text into vectors. Each column of a vector represents a word. The values in each cell of a row show the number of occurrences of a word in a sentence.
  The bag-of-words model is a way of representing text data when modeling text with machine learning algorithms.

  The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modeling and document classification.

  Text data is used in natural language processing (NLP), which interacts between humans and machines using natural language.

  Text reviews provided by the customers are of different lengths. By converting from text to numbers, we can represent a Label by a finite length of the vector. In this way, the length of the vector will be equal for each Label, irrespective of the text length.

- **K-Nearest Neighbour:**

  K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.KNN algorithm stores all the available data and classifies a new data point based on the similarity.

  K in KNN is a parameter that refers to the number of nearest neighbours to a particular data point that are to be included in the decision making process. This is the core deciding factor as the classifier output depends on the class to which the majority of these neighbouring points belongs.

Consider if the value of K is 5, then the algorithm will take into account the five nearest neighbouring data points for determining the class of the object. Choosing the right value of K is termed as Parameter Tuning. As the value of K increases the prediction curve becomes smoother. By default the value of K is 5. There is no structure way to find the value of K, however the optimal value of K is the square root of the total number of samples that are present in the dataset. The value of K is generally taken as an odd value so as to avoid ties during decision making. An error plot or accuracy plot is generally used to find the most appropriate value of K.

- **Decision Tree Classifier:**

  In the Machine Learning world, Decision Trees are a kind of non parametric models, that can be used for both classification and regression.

  This means that Decision trees are flexible models that don't increase their number of parameters as we add more features, and they can either output a categorical prediction or a numerical prediction. They are constructed using two kinds of elements: nodes and branches. At each node, one of the features of our data is evaluated in order to split the observations in the training process or to make an specific data point follow a certain path when making a prediction. When they are being built decision trees are constructed by recursively evaluating different features and using at each node the feature that best splits the data.

  1. The Root Node: In a normal decision tree it evaluates the variable that best splits the data.
  2. Intermediate nodes: These are nodes where variables are evaluated but which are not the final nodes where predictions are made.
  3. Leaf nodes: These are the final nodes of the tree, where the predictions of a category or a numerical value are made.

- **Random Forest Classifier:**

  Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

  A random forest consists of multiple random decision trees. Two types of randomnesses are built into the trees. First, each tree is built on a random sample from the original data. Second, at each tree node, a subset of features are randomly selected to generate the best split.

- **XGBoost Classifier:**

  XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that has gained popularity for its exceptional performance and versatility in various data science projects. It belongs to the family of gradient boosting methods, which are ensemble techniques that combine weak learners (decision trees in the case of XGBoost) to create a strong predictive model.

  Here's a summary of the working of the XGBoost classifier:

  1. Initialization: The XGBoost classifier is initialized with hyperparameters.
  2. Base Model Creation: It starts with a single decision tree as the base model.
  3. Gradient Calculation: XGBoost calculates the gradient of the loss function.
  4. Boosting Iterations: It performs multiple boosting iterations to create new weak learners.
  5. Loss Function Optimization: XGBoost optimizes the loss function by fitting the weak learner to the negative gradient.
  6. Tree Addition and Combination: The new weak learner is added to the ensemble, and predictions are combined.
  7. Regularization: XGBoost applies regularization techniques to control over fitting.
  8. Prediction: The final ensemble model is used to make predictions on new data.
  9. Evaluation: The performance of the classifier is evaluated using appropriate metrics.
  10. Model Tuning: Hyper parameters can be tuned to optimize the model's performance.

- **Light Gradient Boosting Machine:**

  LightGBM is designed to be highly efficient and can handle large-scale datasets with millions of instances and features. It uses a histogram-based approach to binning, which reduces memory usage and speeds up training. Working of the LightGBM (LGBM) classification algorithm is as given

  1. Data Preparation: Prepare the dataset with features and target labels.
  2. Initialization: Initialize the LGBM classifier with hyperparameters.
  3. Dataset Split: Split the dataset into a training set and optionally a validation set.
  4. Base Model Creation: Start with a single decision tree as the base model.
  5. Boosting Iterations: Perform multiple boosting iterations to create new weak learners.
  6. Gradient Calculation: Calculate the gradient and Hessian of the loss function.
  7. Leaf-wise Tree Growth: Use a leaf-wise tree growth strategy for faster training.

8. Loss Function Optimization: Optimize the loss function by fitting the weak learner to the negative gradient.
9. Tree Addition and Combination: Add the new weak learner to the ensemble and combine predictions.
10. Regularization: Apply regularization techniques to control overfitting.
11. Prediction: Use the final ensemble model to make predictions on new data.
12. Evaluation: Evaluate the performance of the classifier using appropriate metrics.
13. Model Tuning: Tune the hyperparameters to optimize model performance.

- **Chi-Square Test of Independence**

  The chi-square test of independence is used to determine if there is a significant association between two categorical variables. The assumption of Chi-Square Test of Independence are as follows:

  1. Independence: The observations should be independent of each other.
  2. Variable type:Both variables are categorical
  3. Sample Size: The sample size should be sufficiently large for reliable results.
  4. Expected frequencies: Expected value of cells should be 5 or greater in at least 80percentage of cells

  The chi-square test statistic is calculated using the following formula:

  $$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

  where:

  1. χ^2 represents the chi-square test statistic, measuring the overall discrepancy between observed and expected frequencies.
  2. O_{ij} refers to the observed frequency in a specific cell of the contingency table.
  3. Eij refers to the expected frequency in a specific cell, assuming independence between the variables.It is calculated as $\frac{rowtotal \times columntotal}{grandtotal}$.

  The test statistic (χ^2) follows a chi-square distribution. The resulting p-value is compared to a chosen significance level (e.g., 0.05) to determine statistical significance. If the p-value is below the significance level, the null hypothesis of independence is rejected, indicating a significant association between the variables.

- **Analysis of variance**

  The One-way Analysis of Variance (ANOVA) is used to determine whether there are any significant differences between the means of three or more independent (unrelated) groups. The One-way ANOVA compares the means between the groups we are interested in and determines whether any of those means are significantly different from each other. Specifically, it tests the null hypothesis:

$$H_0 : \mu_1 = \mu_2 = \ldots = \mu_k$$

where μ = group mean and k = number of groups. It is assumed that the k populations have common variance σ^2. If, however, the One-way ANOVA returns a significant result, we accept the alternative hypothesis H_1, which is that there are at least 2 group means that are significantly different from each other.

- **Box Plot**

  A Box Plot is a convenient way of graphically depicting groups of numerical data through their quartiles. It is used to better understand how values are spaced out in different sets of data. It is used to display the patterns of quantitative data. It may also have lines extending vertically from the boxes indicating variability outside the upper and lower quartiles and the outliers may be plotted as individual points. Box plots are non-parametric. They display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. The spacings between the different parts of the box indicate the degree of dispersion and skewness in the data, and show outliers. Box plot often provides information about the shape of a data set.

- **Kruskal Wallis Test**

  Kruskal Wallis test is a non parametric statistical test used to compare the medians of two or more independent groups. It is an extension of the Mann Whitney U test for more than two groups. The test is used when the assumptions of normality and equal variances required by parametric tests like ANOVA are not met.
  Procedure of Kruskal Wallis Test is as given below

  1. Null hypothesis (H0): The medians of all groups are equal. Alternative hypothesis (HA): At least one group median is different from the others.
  2. Rank the data: Combine all the observations from different groups and rank them from smallest to largest. Assign ranks to each observation, regardless of the group it belongs to.
  3. Calculate the sum of ranks for each group: Add up the ranks for each group separately.
  4. Calculate the test statistic (H): The Kruskal Wallis test statistic (H) is calculated using the ranks. It compares the variability between groups to the variability within groups.

$$H = \frac{12}{N(N+1)} \sum (R_j - \frac{N+1}{2})^2 \div (N-1)$$

where N is the total number of observations and Rj is the sum of ranks for each group.

5. Calculate the degrees of freedom: Degrees of freedom (df) in the Kruskal Wallis test are determined by the number of groups (k) minus 1.

$$df = k - 1$$

6. Determine the critical value and compare with the test statistic: The critical value for the KruskalWallis test depends on the chosen significance level (α) and the degrees of freedom. It can be obtained from a chisquare distribution table or using statistical software. If the calculated test statistic (H) exceeds the critical value, you reject the null hypothesis, indicating that at least one group median is significantly different.

- **Cramer V test**

  The Cramer's V test is a statistical measure used to assess the strength and association between categorical variables. It is particularly useful when analyzing the relationship between two nominal variables. The test produces a value ranging from 0 to 1, with higher values indicating a stronger association.

  The formula for Cramer's V is derived from the chi-square statistic and the number of observations. It is calculated as follows:

$$V = \sqrt{\left(\frac{\chi^2}{n}\right)\left(\frac{\min(r-1, c-1)}{n}\right)}$$

  Where:
  V is the Cramer's V statistic
  χ^2 is the chi-square statistic
  n is the total number of observations
  r is the number of rows in the contingency table
  c is the number of columns in the contingency table

  To interpret the Cramer's V value:
  A value close to 0 suggests a weak association between the variables.
  A value around 0.5 indicates a moderate association.
  A value close to 1 represents a strong association.

3 Chapter 3
RESULTS AND DISCUSSION

3.1 Graphical Analysis

In this section we will carry out analysis of various facotrs using graphical methods.

3.1.1 Analysing target variable

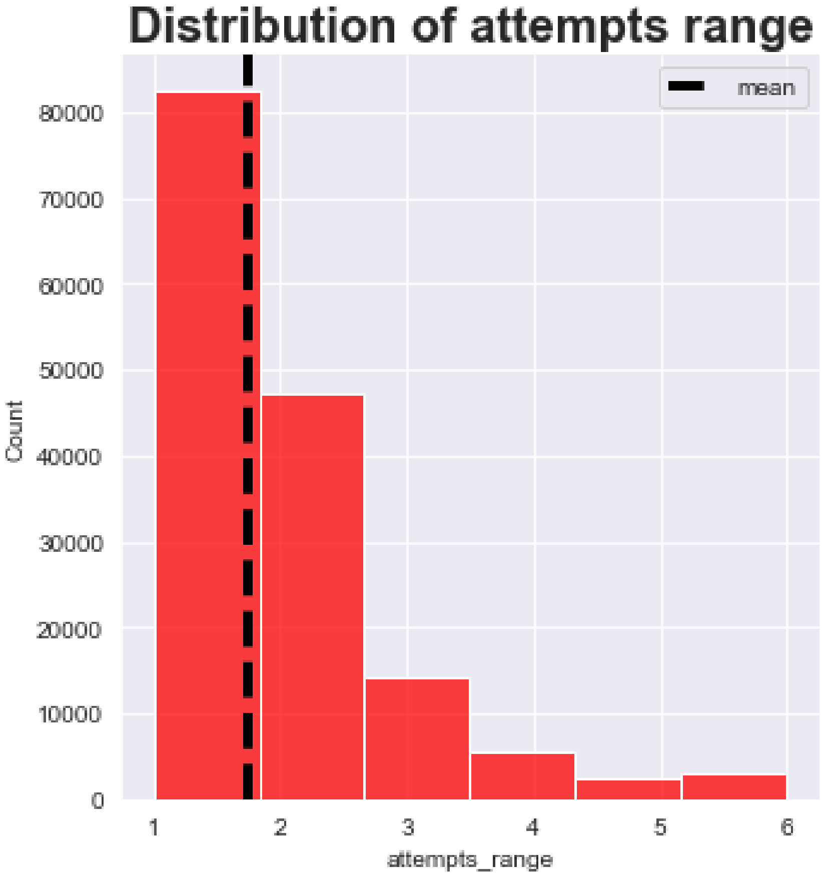


Figure 1: Distribution of target Variable

By looking at distribution plot of attempts range we can observe that attempts range is not normally distributed its right skewed in nature.Attempts range 1 has highest no of observation followed by attempts range 2.Where as attempts range 5 has least no of observation among all the 6 attempts ranges.

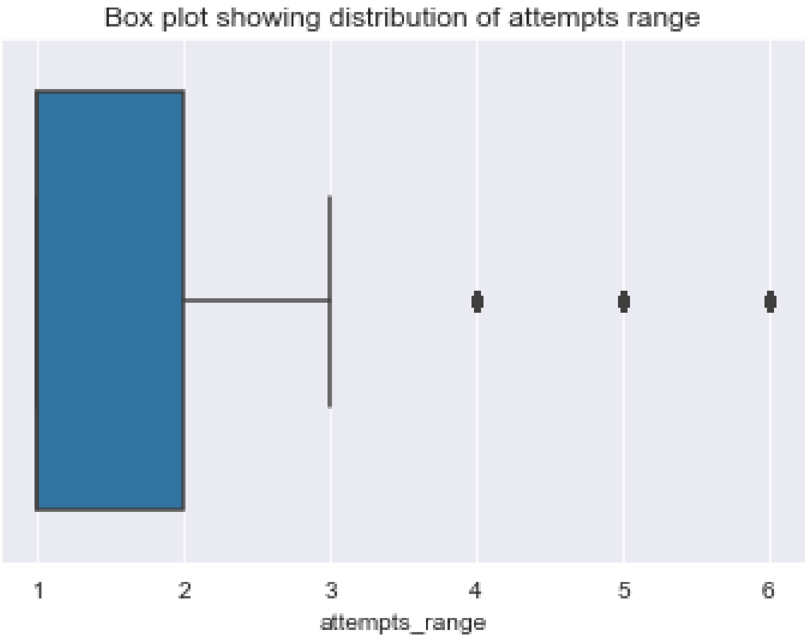


Figure 2: Box plot of target variable

From the above plot we can see that attempt range is highly dense around 1 and 2.Attempt range three has really very less observation as compared to 2 and 1 but it has decent amount of observation as compared to 4,5,6. Attempt range 1 has high density may be its because the problems are easier one to solve or expert and advanced people may solve with less attempts because of experience and people who are beginner and intermediate are may be getting easy basic level problem.

3.1.2 Distribution of submission count

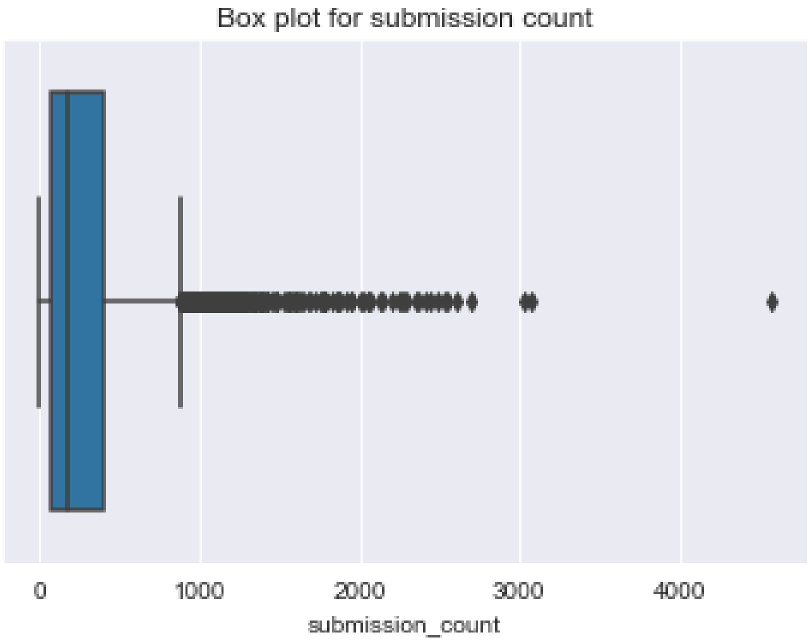


Figure 3: Box plot of submission count

Submission count is highly densed between 50-450 It looks like there are few outliers . Lower quartile is at 71 and third quartile is at 393. Average submission count is 303 with median of 171.

3.1.3 Analysing variable contribution

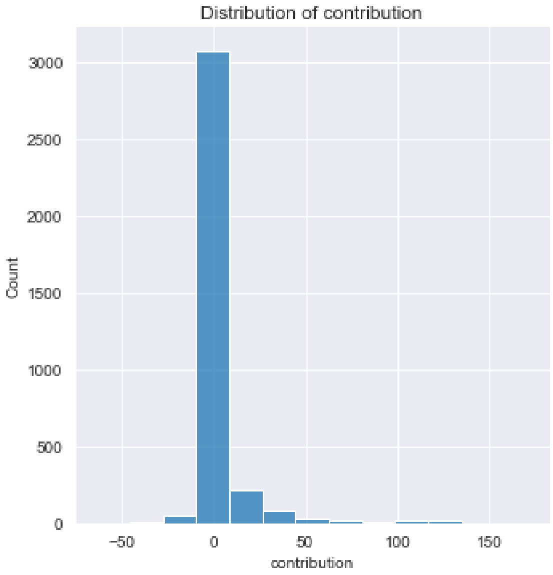


Figure 4: Histogram showing submission count

It looks like there are lots of zero entries and there some negative values.This variable requires some extra analysis in future before involving passing it into model.

3.1.4 Checking relation between user rank and contribution

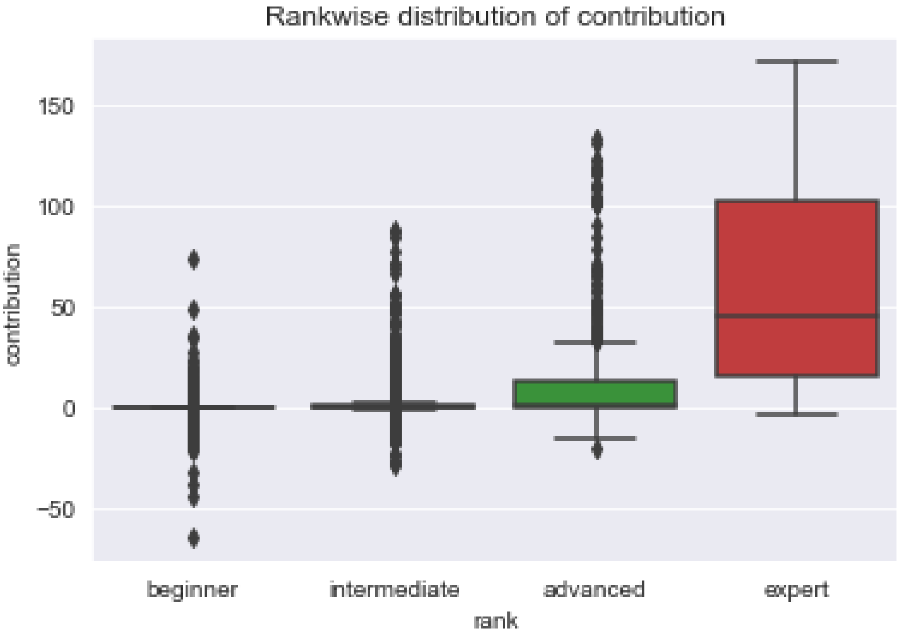


Figure 5: Box plot of contribution based on user rank

By looking at distribution of contribution of users based on user rank we can observe that contribution are majorly made by experts and advanced user.Contribution by users of intermediate and beginner level is very small, it may be due lack of expertise or domain knowledge.

Table 2: Table showing rank wise user count

User Rank	Number of Users
Beginner	53044
Intermediate	66300
Expert	5209
Advanced	30122

But previously we have observed that there are lots of zero values in contribution columns.Further looking at no.of users per rank majority of users belongs to beginner and intermediate level, may be that is the reason to get many zero entries in contribution column.

3.1.5 Comparing Attempts range with user rank

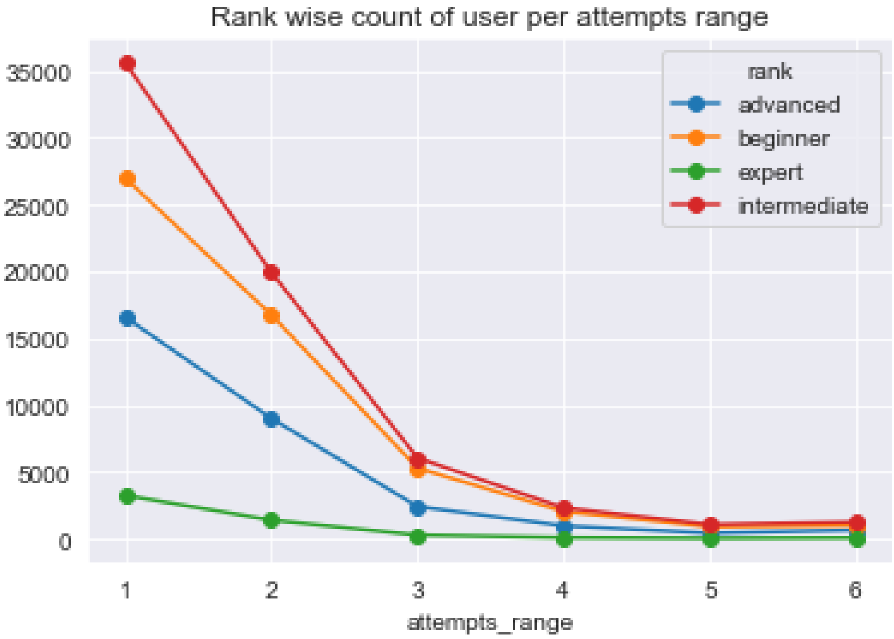


Figure 6: Line plot of rank wise count of user per attempts range

While studying target variable we had seen that attempt range is highly distributed around 1 and 2. Here we observe that even most of the beginner and expert takes 1 or 2 attempts to solve the problem. The reason why many beginner and intermediate solve problem in one attempt may be it is because that problems given to them are may be simpler one to solve.

Table 3: Table showing user count per rank

User Rank	Number of Users
Beginner	53044
Intermediate	66300
Expert	5209
Advanced	30122

In the above plot very few expert were taking 1 and 2 attempts range.But now looking at cross table there are very few users at expert level. May be this is why graph is showing very few people from expert level who solved problem within 1 or 2 attempts range.

3.1.6 Top 10 countries with highest users

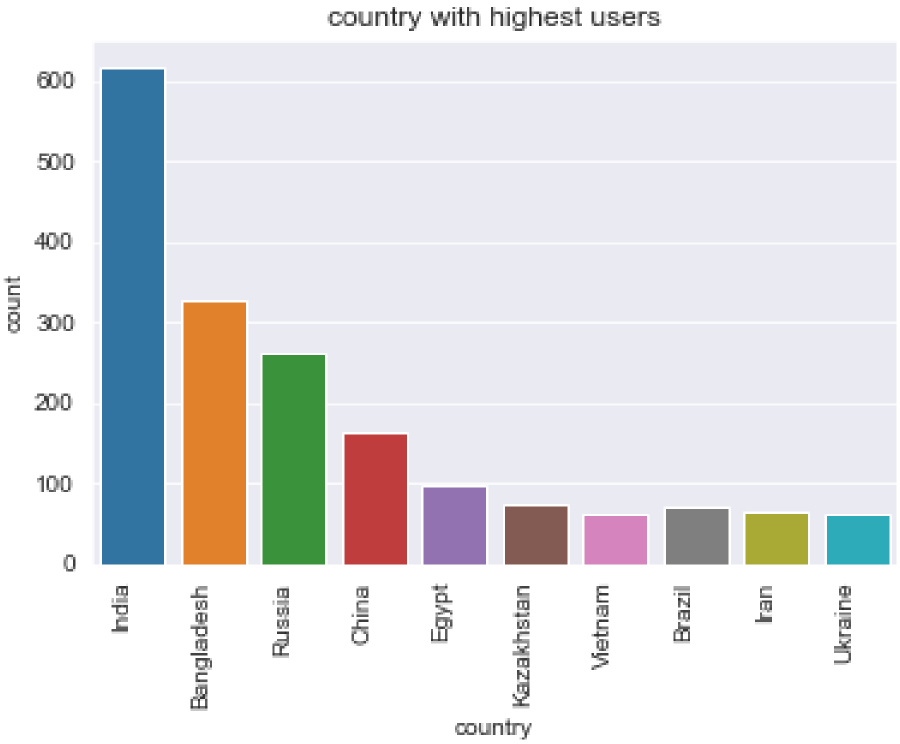


Figure 7: Bar graph showing country wise user count

India has highest users followed by Bangladesh and Russia where as China and Egypt stands at 4th and 5th place based on no of users. Countries like India , Bangladesh and China which are three among 5 countries with highest users belongs to Asia continent.

3.1.7 Time spent by users of different rank

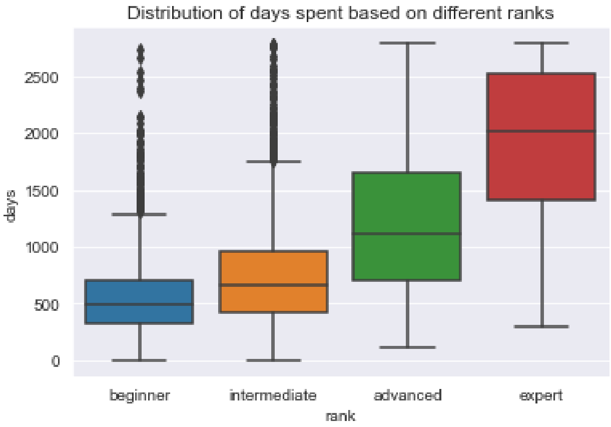


Figure 8: Box plot of days based on user rank

By looking at distribution of time spent(i.e no of days) based on user levels , experts have spent on a median of 2000 days.Where as intermediates spent 600 to 700 days. There are some outlier in no of days spent for beginner and intermediate level users.

3.1.8 Problems type

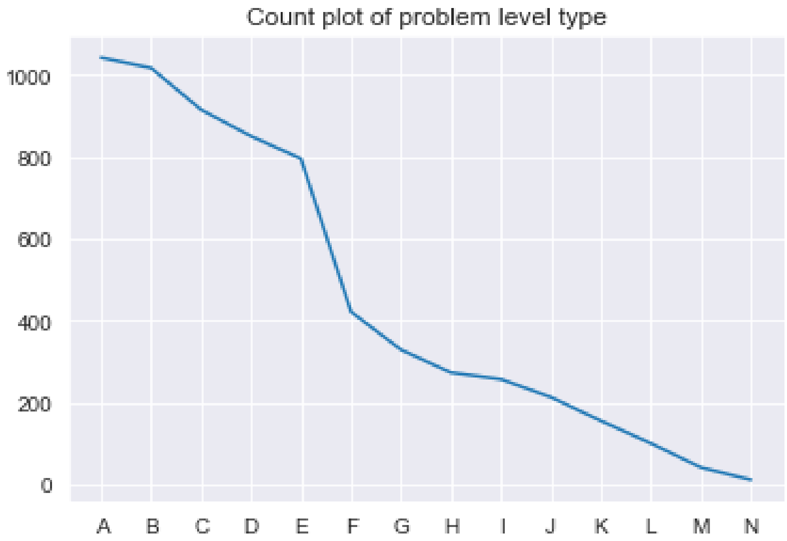


Figure 9: line graph showing number of problem per problem type

Problem of level type A has highest no of observation i.e, 60000 .Problems of level type between E to N has very few observations.

3.1.9 Distribution of ratings

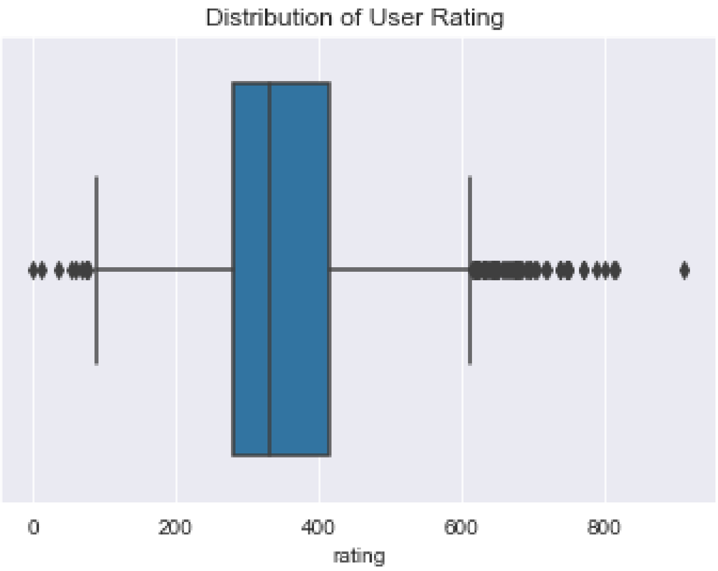


Figure 10: Box plot of ratings

By looking at box plot we can say that min of user rating is 0.0 and max of user rating is 911.124. 25% of rating is below 279 and 75% percent of ratings is less than 413.4175 . Average user rating is 350.2,with median rating of 330.

3.1.10 Analysing no of problem solved based on user rank

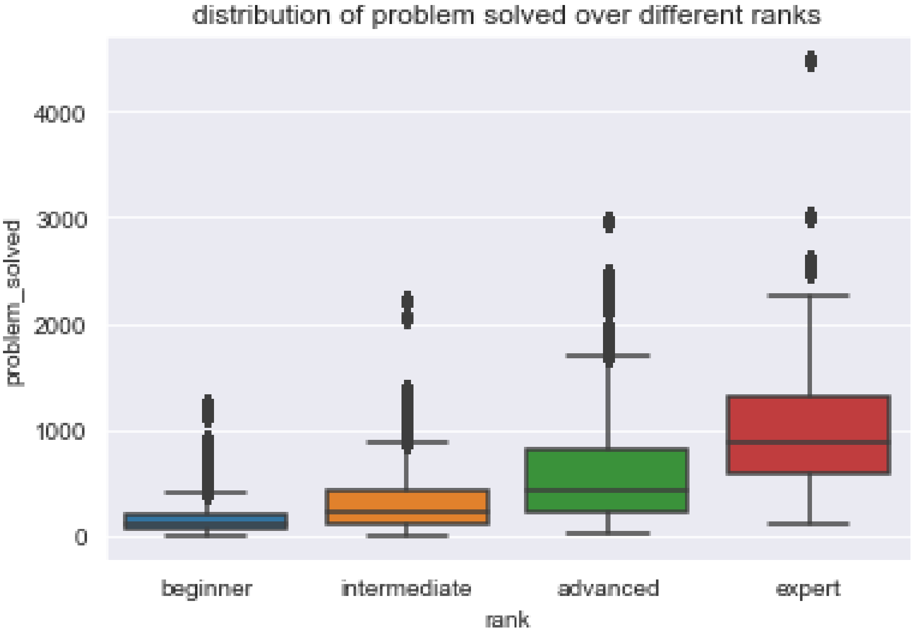


Figure 11: Box plot of number of problems solved.

Experts are found to be solving more no of problem .Distribution of problem solved found to be very less in beginner and intermediate.Suggesting appropriate problem is very important to keep beginners engaged in website and to retain their interest.

3.1.11 Conclusion

1. Attempts range is highly densed between 1 and 2 and there are few users who took 5 attempts to solve problems.
2. Most of the users has submission counts between the range 50 to 450. 25% of observations in submission counts lies below 71.Average submission count is 303 with median value 171.
3. Majority of the contribution are made by users belonging to expert categories and contribution made by users from beginner and intermediate category is very low.
4. There are lots of zero entries in contribution column.Number of users from beginner category is really high and contribution from them is really low may be this is one of the reason why contribution column has too many zero entries.

5. Most of the users from Advanced and Expert category takes one or two attempts. But even users from beginner and intermediate level are able to solve problems within one or two attempts range.
6. Most of the users belongs to India, Bangladesh, Russia, China and Egypt. Three countries in top 5 countries with highest users belongs to Asian continent.
7. Users from expert category are found to be spending more time in website followed by users from advanced category.Experts have spent median of 2000 days in the website.
8. Problems of type A, B, C, D, E, F have large no of observation, where as problems of type M and N have very hand full of observations.
9. 25% of ratings are below 279 and 75% of ratings are below 413.41 , average user rating is 350.2 .
10. Users belonging to expert level are found to be solving more no of problems followed by users from advanced category.It's obvious that more experience you gain as you solve more and more problems.

3.2 Statistical Testing

In this section we try to study the relation between different factors using various parametric and nonparametric statistical techniques.

3.2.1 To study the effect of various factors in dataset on attempts range using ANOVA

To check whether number of problem solved affects the attempts_range.

Testing the assumptions of ANOVA.

Since the number.of observations are large we assume that observations follow normal distribution.

Bartlett test for checking the homogeneity of variance.

Null hypothesis H_0: The variance of the data within each group categorised by attempts range are equal.

Alternative hypothesis H_1: The variance of the data within each group categorised by attempts range are not equal.

The values obtained as follows:
Test statistics: 603.2943
P value: 3.9273e-128

Here p value is less than 0.05 so we reject the null hypothesis and conclude that the variance of at least one group is significantly different from the others.

For ANOVA assumption of homogeneity of variance fails.So we use alternative non parametric Kruskal wallis H test.

Null hypothesis H_0:The median number of problems solved among all groups,categorized by attempts range are equal.

Alternative hypothesis H_1: The median number of problems solved are not equal among all groups,categorized by attempts range.

The values obtained as follows
Test statistic = 84.8123
p value = 8.2418e-17
Here p value is less than 0.05 so we reject the null hypothesis and conclude that there is a significant difference in median no of problem solved , between at least two groups of attempts range.

Finally we conclude that there is a significant affect of no of problem solved on attempt range.

To test whether number of days spent has any significant affect on attempts_range.

Testing the assumptions of ANOVA.

Since the number of observations are large we assume that observations follow normal distribution.

Bartlett test for checking the homogeneity of variance.

Null hypothesis H_0: The variance of the data within each group categorised by attempts range are equal.

Alternative hypothesis H_1: The variance of the data within each group categorised by attempts range are not equal.

The values obtained as follows
Test statistic=756.4298
p value=3.0765e-161

Here p value is less than 0.05 so we reject the null hypothesis and conclude that the variance of at least one group is significantly different from the others.

For ANOVA assumption of homogeneity of variance fails.So we use alternative non parametric Kruskal Wallis H test.

Null hypothesis H_0:The median number of days spent in website among all groups,categorized by attempts range are equal.

Alternative hypothesis H_1: Median number of days spent in website are not equal among all the groups of attempts range.

The values obtained as follows
Test statistic=272.4311
p value=8.4103e-57
Here p value is less than 0.05 so we reject the null hypothesis and conclude that there is a significant difference in median no of days spent,between at least two groups of attempts range.

Finally we conclude that on attempt range there is a significant affect of number of days spent by user in website.

To test whether follower count has any significant affects on attempts_range.

Testing the assumptions of ANOVA.

Since the no.of observations are large we assume that observations follow normal distribution.

Bartlett test for checking the homogeneity of variance.

Null hypothesis H_0: The variance of the data within each group categorised by attempts range are equal.

Alternative hypothesis H_1: The variance of the data within each group categorised by attempts range are not equal.

The values obtained as follows
Test statistic : 8490.8564
p value : 0.0

Here p value is less than 0.05 so we reject the null hypothesis and conclude that the variance of at least one group is significantly different from the others.

For ANOVA assumption of homogeneity of variance fails.So we use alternative non parametric Kruskal wallis H test.

Null hypothesis H_0:The median number of follower counts among all groups,categorized by attempts range are equal.

Alternative hypothesis H_1: There is a significant difference in median no of follower counts,between at least two groups of attempts range.

The values obtained as follows
Test statistic : 106.7933
p value : 1.9494e-21
Here p value is less than 0.05 so we reject the null hypothesis and conclude that there is a significant difference in median no of follower counts,between at least two groups of attempts range.

Finally we conclude that there is a significant affect of follower count on attempts range .

3.2.2 To study the association between various factors in data set and attempts range using χ^2 Test

To determine whether user rank affects the attempts range using χ^2

The following table shows the no of observations for different attempts range grouped by user rank.

Table 4: Cross table for user rank and attempts range

User_level:Attempts Range	**1**	**2**	**3**	**4**	**5**	**6**
Advanced	16602	9017	2426	982	435	660
Beginner	26991	16768	5303	2058	912	1012
Expert	3263	1408	290	105	58	85
Intermediate	35620	19942	6063	2333	1079	1263

Null hypothesis H_0: There is no association between user rank and attempts range.

Alternative hypothesis H_1: There is an association between user rank and attempts range.

The values are as follows
Test statistics :456.1746
p value :1.0256e-87
Degrees of freedom : 15
Here p value is less than 0.05 so we reject the null hypothesis and conclude that user rank and attempts range are associated.

Determining the strength of association between user rank and attempts range using Cramer's V test

Cramer V value : 0.0314
Cramer V value is 0.0314 which is very low .So we conclude that there is a weak relationship between user rank and attempts_range.

3.2.3 To study the association between various factors in dataset and attempts range using ordinal regression.

To study the association between submission count and attempts range.

H_0 : There is no association between submission count and attempts range.
H_1 : There is a association between submission count and attempts range.

The values are as follows
p value : 0.0

Coefficient for submission counts : -3.494e-05
Odds ratio : 0.9994
Here p value is 0.0 which is less than 0.05 so we reject the null hypothesis and conclude that there exist a significant association between submission count and attempts range.
The negative coefficient suggests that as submission count increases the chances of moving to a higher value in the ”attempts range” decreases.
The odds ratio of 0.99996506 indicates that for every one unit increase in ”submission count,” the odds of moving to a higher category of ”attempts range” decrease by a factor of approximately 0.99996506.

To study the association between ratings and attempts range.

H_0 : There is no association between user ratings and attempts range.
H_1 : There exist an association between user rating and attempts range.
The values are as follows:
Coefficient for rating : -0.0005
p value : 0.0
Odds ratio : 0.9995
Here p value is 0.0 which is less than 0.05 so we reject the null hypothesis and conclude that there exist a significant association between user rating and attempts range.
The negative coefficient suggests that as value of ratings increases, chances of moving to a higher category in the attempts range becomes less likely .
The odds ratio of 0.9995 indicates that for every one unit increase in the rating variable, the log odds of moving to a higher category of attempts range decrease by a factor of approximately 0.9995.

To study the association between points and attempts range.

H_0 : There is no association between points and attempts range.
H_1 : There exist an association between points and attempts range.

The values are as follows:
Coefficient for rating : 0.0005
p value : 0.0
Odds ratio : 1.0005
Here p value is 0.0 which is less than 0.05 so we reject the null hypothesis and conclude that there exist a significant association between variable points and attempts range.
The positive coefficient suggests that increase in ”points” is associated with a higher chances of moving to a higher category of ”attempts range.

To study the association between maximum ratings and attempts range.

H_0 : There is no association between maximum ratings and attempts range.
H_1 : There exist an association between maximum ratings and attempts range.

The values are as follows:
Coefficient for maximum rating : -0.0006
p value : 0.0
Odds ratio : 0.9994
Here p value is 0.0 which is less than 0.05 so we reject the null hypothesis and conclude that there exist a significant association between variable maximum ratings and attempts range.
The negative coefficient suggests that as value of maximum ratings increases, chances of moving to a higher category in the attempts range becomes less likely. The odds ratio of 0.9994 indicates that for every one unit increase in ”max rating,” the odds of moving to a higher category of ”attempts range” decrease by a factor of approximately 0.9994.

3.2.4 Conclusion

1. There is a significant effect of number of problems solved on attempts range.
2. From Kruskal Wallis test, there exist an association between number of days spent and attempts range.
3. There is a significant effect of followers count on attempts range.
4. There exist a significant association between no of days spent and attempts range.
5. From Chisquare test of independence and Cramer v test it is found that there exist an weak association between user rank and attempts range.
6. There is a association between attempts range and submission count.As submission count increases the chances of moving to a higher value in the ”attempts range” decreases.
7. As value of ratings increases, chances of moving to a higher category in the attempts range becomes less likely.
8. Increase in ”points” is associated with a higher chances of moving to a higher category of ”attempts range. As points increases attempts range also increases.
9. As value of maximum ratings increases, chances of moving to a higher category in the attempts range decreases.

3.3 Machine Learning Models

In this section we discuss about various machine learning that built to predict the attempts range.

3.3.1 Data Splitting

The data is splitted into train and test in the ratio 75:25. The shape of the data for train and test after splitting is as shown below :
x train shape:(116006, 10)
y train shape:(116006,)
x test shape:(38669, 10)
y test shape : (38669,)
Following featurizations are applied only to the train data, since test data is considered as unseen data. Test data will be transformed using the featurization objects of train data.

3.3.2 Data transformation

Attempts range is the target variable which need to be predicted.Below transformation should be performed before fitting the model.

1. Standardising numeric data.
2. Encoding categorical data.
3. Converting text to vector by Bag Of Word.

After the above steps, shape of the data is as given below
Train data :(116006 , 49) (116006,1)
Test data : (38669,49) (38669, 1)

3.3.3 Model Building

Logistic Regression (Multiclass)

From the model below observations are found
F1 score of train data :0.5348
F1 score of test data : 0.5332
Test accuracy of the model : 53.3%
Train accuracy of the model : 53.48%

Figure 12: Confusion matrix for Logistic regression model(Multiclass)

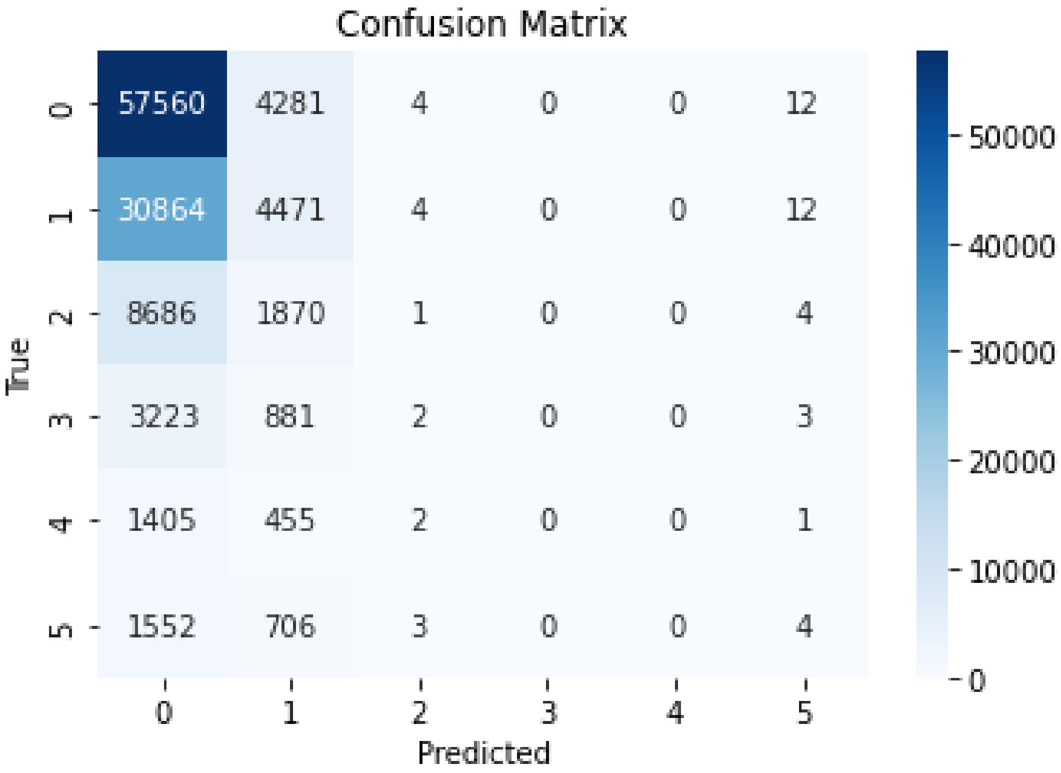


Table 5: Classification Report of Logistic Regression model

Class	Precision	Recall	F1-score	Support
0.0	0.56	0.93	0.70	20619
1.0	0.34	0.12	0.18	11784
2.0	0.00	0.00	0.00	3521
3.0	0.00	0.00	0.00	1369
4.0	0.00	0.00	0.00	621
5.0	0.08	0.00	0.00	755
Accuracy			0.53	38669
Macro avg	0.16	0.18	0.15	38669
Weighted avg	0.40	0.53	0.43	38669

Here both test and train accuracy are low and we can observe that f1 score for classes other than 0 and 1 are zero so we try some more models.

Decision Tree Classifier

From the model below observations are found

F1 score of train data : 0.9477

F1 score of test data : 0.5574

Train accuracy of the model : 95%

Test accuracy of the model : 56%

Figure 13: Confusion matrix for Decision Tree Classifier

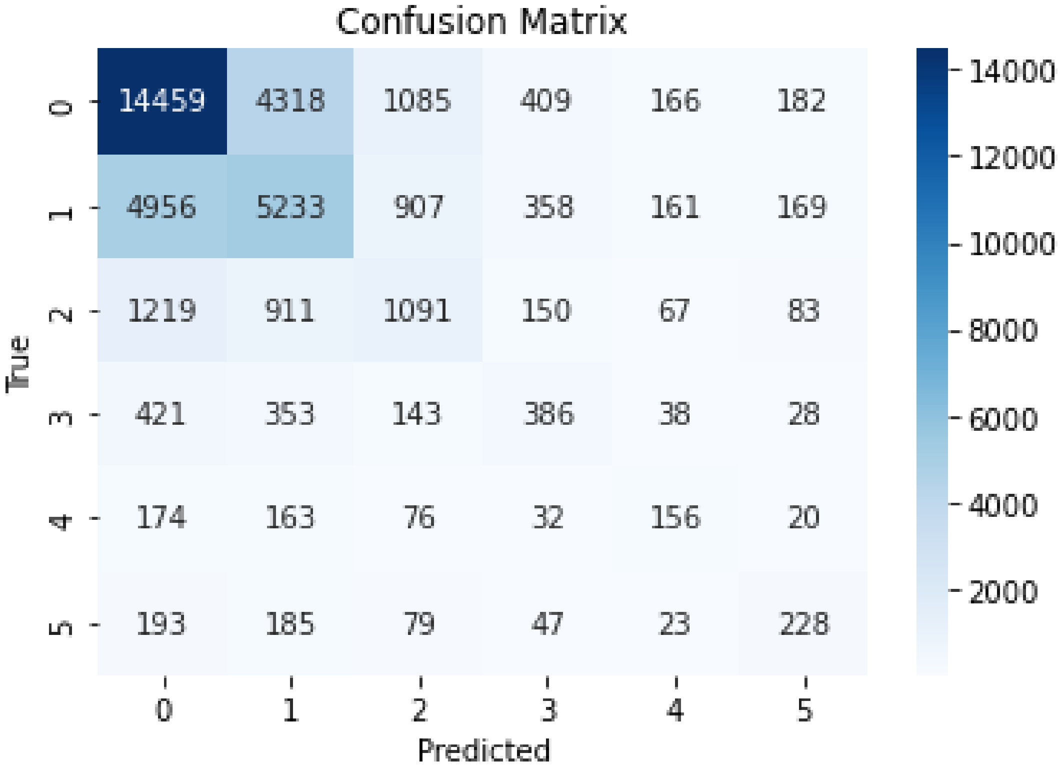


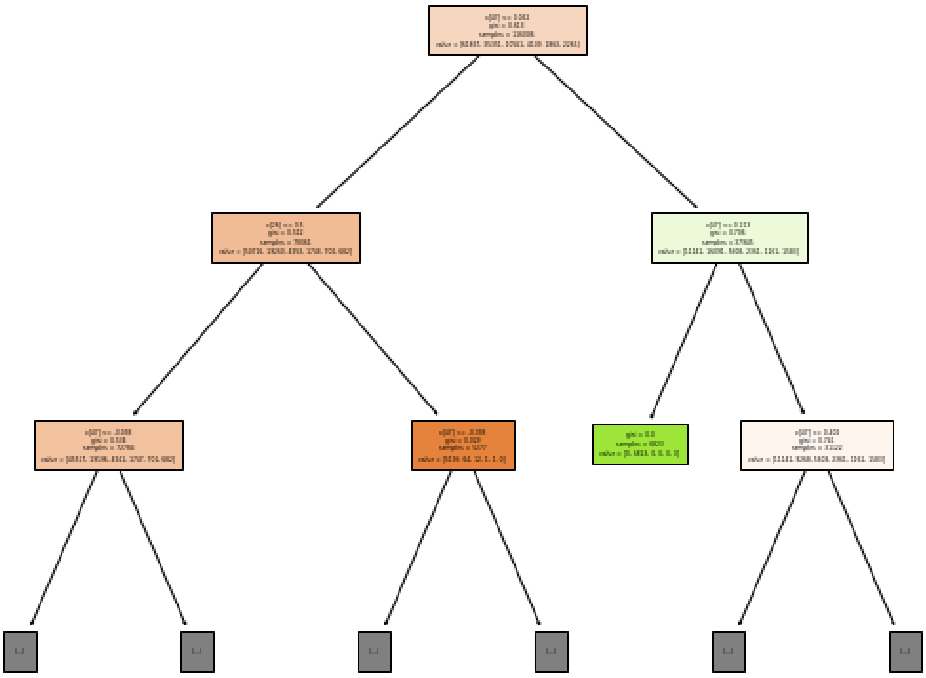
Table 6: Classification Report of Decision Tree Classifier

Class	Precision	Recall	F1-score	Support
0.0	0.67	0.70	0.69	20619
1.0	0.47	0.44	0.46	11784
2.0	0.32	0.31	0.32	3521
3.0	0.28	0.28	0.28	1369
4.0	0.26	0.25	0.25	621
5.0	0.32	0.30	0.31	755
Accuracy			0.56	38669
Macro avg	0.39	0.38	0.38	38669
Weighted avg	0.55	0.56	0.55	38669

Here accuracy for training data set is high(95%) and for test data accuracy is low (56%)

there is a problem of over fitting.

Figure 14: Tree Structure of Decision Tree Classifier



Random Forest Classifier

From the model below observations are found

F1 score of train data : 0.9477

F1 score of test data : 0.6055

Train accuracy of the model : 95%

Test accuracy of the model : 61%

Figure 15: Confusion matrix for Random Forest Classifier

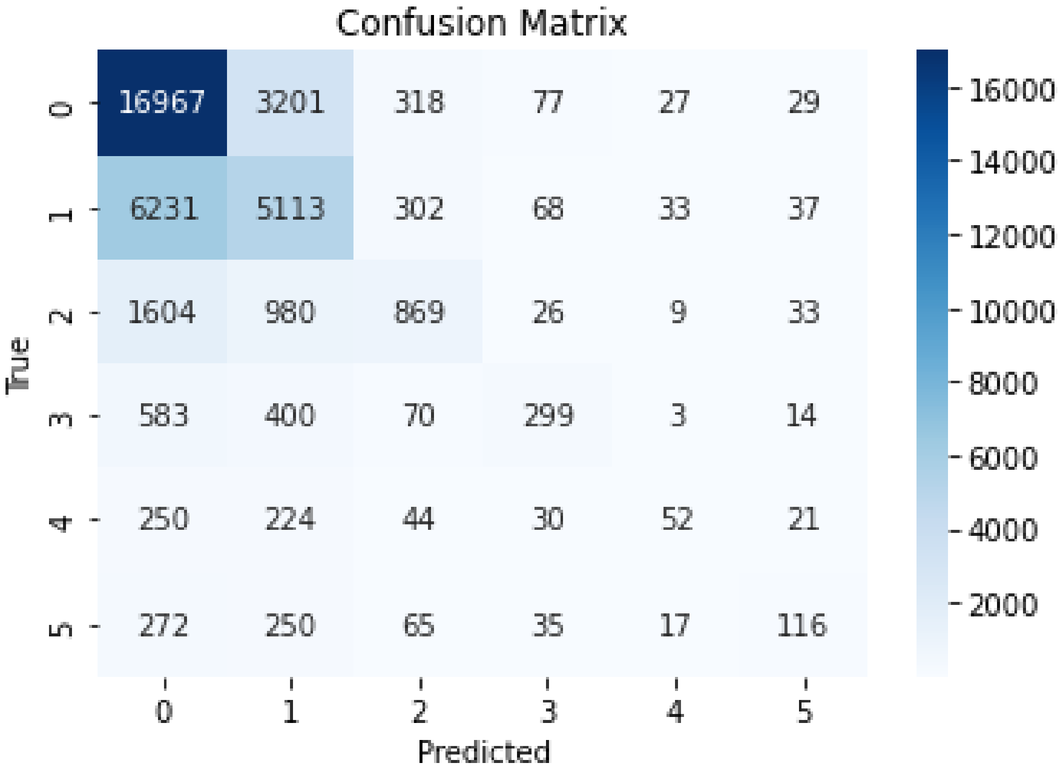


Table 7: Classification Report of Random Forest Classifier

Class	Precision	Recall	F1-score	Support
0.0	0.65	0.82	0.73	20619
1.0	0.50	0.43	0.47	11784
2.0	0.52	0.25	0.33	3521
3.0	0.56	0.22	0.31	1369
4.0	0.37	0.08	0.14	621
5.0	0.46	0.15	0.23	755
Accuracy			0.61	38669
Macro avg	0.51	0.33	0.37	38669
Weighted avg	0.58	0.61	0.58	38669

Even in this model we are facing the problem of over fitting. In this model there is drop in accuracy for class 4 and 5 as compared to Decision Tree classifier but over all accuracy is good as compared to Decision Tree Classifier.

We tried tuning hyper parameters using randomised search method but accuracy fell down to 53%.

When we applied Random Forest Classifier with re-assigned class weights (based no of observation that classes has in training set) we got the following observation:
F1 score of train data : 0.9392
F1 score of test data :0.5937
Train accuracy of the model : 94%
Test accuracy of the model : 59%

Table 8: Classification Report Random Forest with reassigned weights

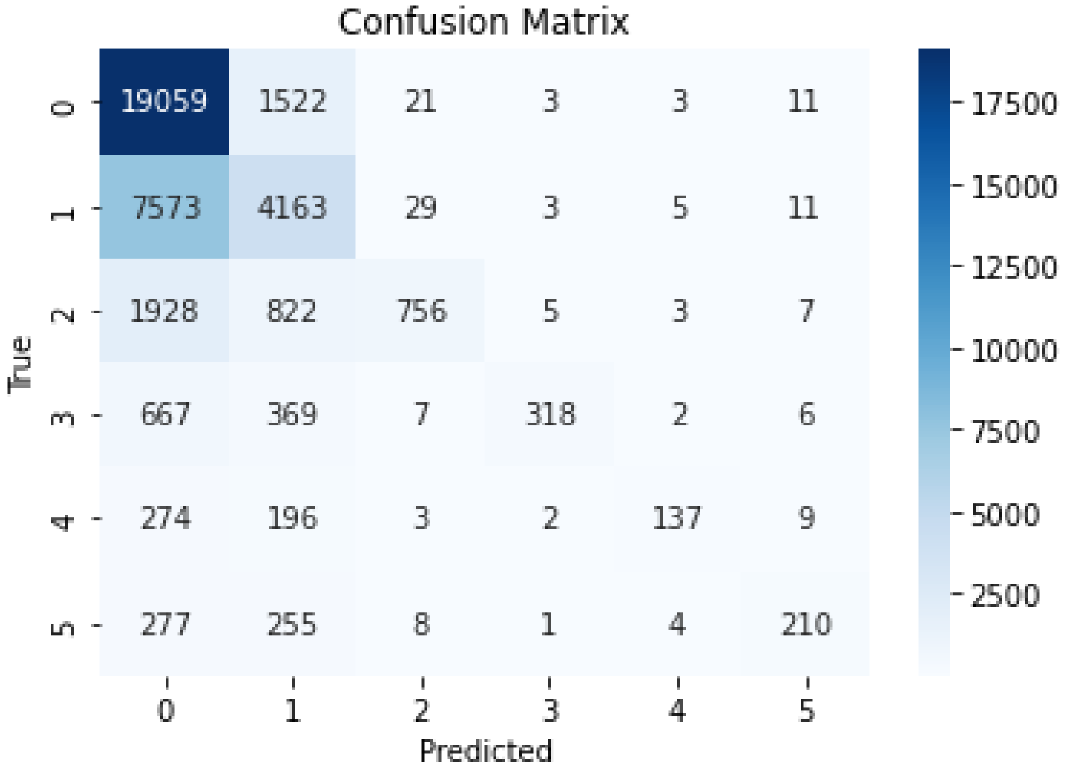
Class	Precision	Recall	F1-score	Support
0.0	0.65	0.80	0.72	20619
1.0	0.50	0.43	0.46	11784
2.0	0.45	0.24	0.32	3521
3.0	0.47	0.25	0.32	1369
4.0	0.43	0.20	0.27	621
5.0	0.48	0.26	0.34	755
Accuracy			0.59	38669
Macro avg	0.50	0.36	0.40	38669
Weighted avg	0.57	0.59	0.57	38669

There is slight improvement in accuracy for each classes.But there is no much changes in over all accuracy.There is improvement in accuracy for classes 3,4,5.

XG_Boost Classifier

From the model below observations are found
F1 score of train data :0.6757
F1 score of test data :0.6373
Train accuracy of the model : 68%
Test accuracy of the model : 64%

Figure 16: Confusion matrix for XG_Boost Classifier



True \ Predicted	0	1	2	3	4	5
0	19059	1522	21	3	3	11
1	7573	4163	29	3	5	11
2	1928	822	756	5	3	7
3	667	369	7	318	2	6
4	274	196	3	2	137	9
5	277	255	8	1	4	210

Table 9: Classification Report of XG_Boost Classifier

Class	Precision	Recall	F1-score	Support
0.0	0.64	0.92	0.76	20619
1.0	0.57	0.35	0.44	11784
2.0	0.92	0.21	0.35	3521
3.0	0.96	0.23	0.37	1369
4.0	0.89	0.22	0.35	621
5.0	0.83	0.28	0.42	755
Accuracy			0.64	38669
Macro avg	0.80	0.37	0.45	38669
Weighted avg	0.66	0.64	0.59	38669

Here we are getting some good improvement in Test accuracy.Even accuracy for each class is looking good.

Light Gradient Boosted Machine

From the model below observations are found

F1 score of train data :0.6583

F1 score of test data : 0.6353

Train accuracy of the model : 66%

Test accuracy of the model : 64%

Figure 17: Confusion matrix for LGBM Classifier

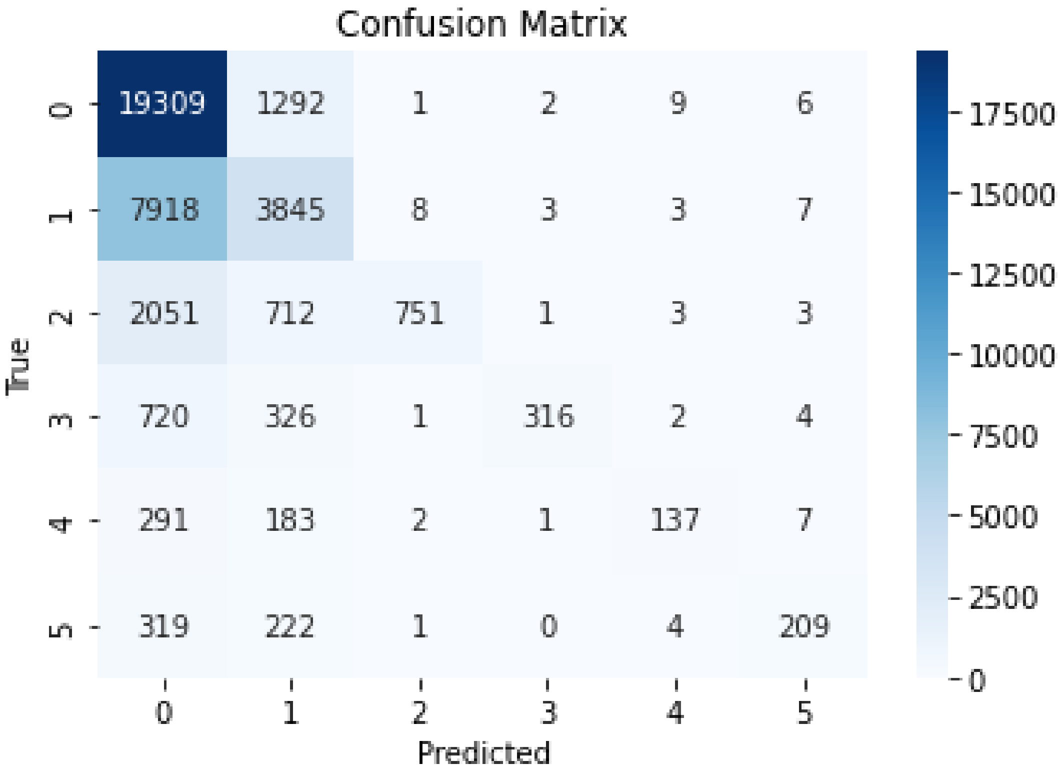


Table 10: Classification Report of LGBM Model

Class	Precision	Recall	F1-score	Support
0.0	0.63	0.94	0.75	20619
1.0	0.58	0.33	0.42	11784
2.0	0.98	0.21	0.35	3521
3.0	0.98	0.23	0.37	1369
4.0	0.87	0.22	0.35	621
5.0	0.89	0.28	0.42	755
Accuracy			0.64	38669
Macro avg	0.82	0.37	0.45	38669
Weighted avg	0.67	0.64	0.59	38669

Test accuracy found to be 64% and train accuracy found to be 66% comparatively XG-Boost is better than this model.

Auto ML

We tried running Auto ML provided by Microsoft with different training time (5 minute , 10 minute, 30 minute, 1 hour, 2 hour) we found the best model with time budget of 1 hour.

Selected model is XGBoost

F1 score of train data :0.6709

F1 score of test data : 0.6393

Train accuracy of the model : 67%

Test accuracy of the model : 64%

Figure 18: Confusion matrix for Auto ML model

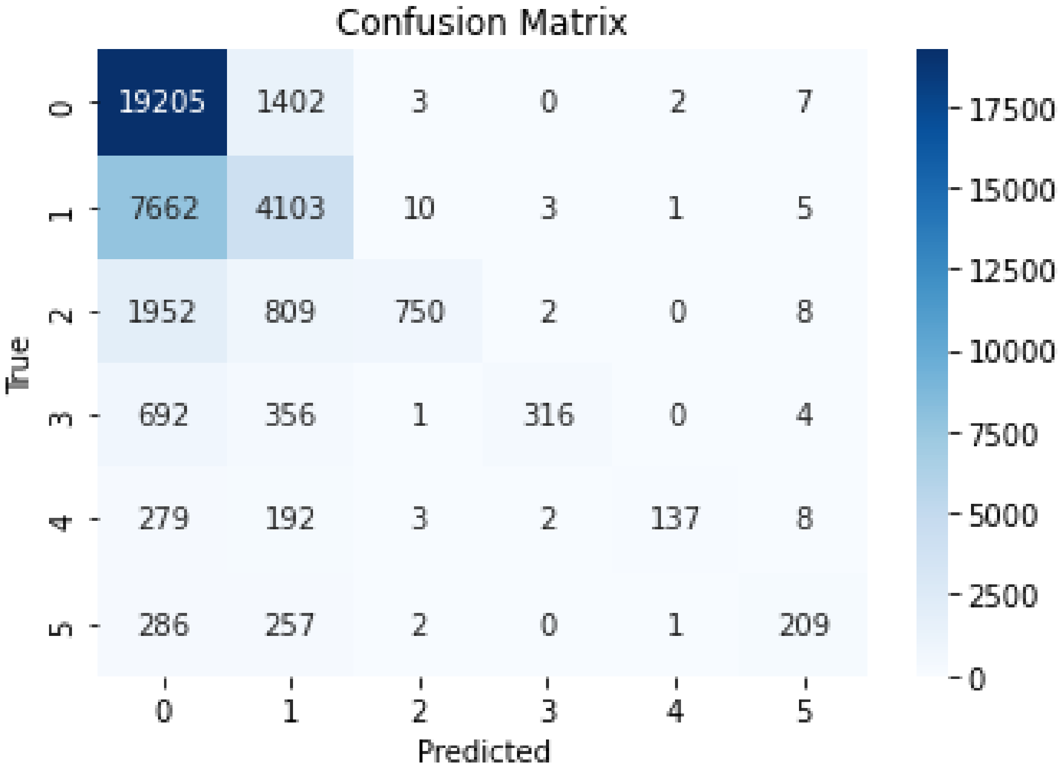


Table 11: Classification Report of Auto ML

Class	Precision	Recall	F1-score	Support
0.0	0.64	0.93	0.76	20619
1.0	0.58	0.35	0.43	11784
2.0	0.98	0.21	0.35	3521
3.0	0.98	0.23	0.37	1369
4.0	0.97	0.22	0.36	621
5.0	0.87	0.28	0.42	755
Accuracy			0.64	38669
Macro avg	0.83	0.37	0.45	38669
Weighted avg	0.67	0.64	0.60	38669

In this model test accuracy is 64% and there is no noticable difference between XGboost model obtained from this model and XGBoost model that one we fitted earlier.

We did check for variance inflation factor and found that there is some relation between maximum rating and followers so we even tried building models by dropping them but there is no any improvement in accuracy.

Up sampling

In this section we tried to improve accuracy by fitting various models on up sampled data (Up sampling is done by using up sampling feature provided by scikit learn).

Random Forest Classifier

F1 score of train data :0.9709
F1 score of test data : 0.58
Train accuracy of the model : 97%
Test accuracy of the model : 58%

Table 12: Classification Report of Random Forest classifier for upsampled data

Class	Precision	Recall	F1-score	Support
0.0	0.65	0.82	0.72	20619
1.0	0.56	0.35	0.43	11784
2.0	0.35	0.31	0.33	3521
3.0	0.34	0.27	0.30	1369
4.0	0.30	0.22	0.25	621
5.0	0.40	0.32	0.36	755
Accuracy			0.59	38669
Macro avg	0.43	0.38	0.40	38669
Weighted avg	0.57	0.59	0.57	38669

There is drop in test accuracy as compared to previous Random Forest model.Also we are not getting any additional improvement from this model fitted on up_sampled data.

XG Boost

F1 score of train data : 0.68
F1 score of test data : 0.5603
There is drastic decrease in accuracy as compared to the previous XGBoost model which was built on original data.

Light Gradient Boosted Machine

F1 score of train data : 0.62
F1 score of test data : 0.561
Even LGBM is not performing good there is drop in its performance after upsampling.

We even tried AUTO ML with training duration of 1 hour and got the accuracy of 0.58 (f1_score micro) which is not that great.

3.3.4 Model comparison

Below table shows the different classification models considered and the corresponding outputs.

Table 13: Model Comparison

Model	Accuracy	F1-score
Random Forest	61%	0.6055
XGBoost	64%	0.6373
LGBM	64%	0.6353
AUTOML	64%	0.6393

3.3.5 Conclusion

From above comparison (Table 13 : Model Comparison) XGBoost , LGBM and model from AUTOML(XGB) are almost similar in accuracy but keeping in mind about accuracy for each categories of target variable we go with XGBoost model for current scenario. Working of Random Forest as well as Decision tree is not that bad but there is a problem of over fitting. They are able to learn train data really well but performance on test data is not so good.But in future with more and well balanced data they may provide better performance.But for right now XGBoost built on default data set is preferred.

4 Chapter 4 Conclusion

To sum up, the following conclusions can be drawn.

1. There is a significant association between attempts taken to solve problems and no of problem solved.
2. There is a weaker association between user rank and the attempts range taken to solve problems.
3. As the value of ratings increases, chances of moving to a higher category in the attempts range become less likely. The odds ratio of 0.9995 indicates that for every one unit increase in the rating variable, the log odds of moving to a higher category of attempts decrease by a factor of approximately 0.9995.
4. There is a significant association between variable points and attempts range.An increase in points is associated with higher chances of moving to a higher category of attempts range.
5. We have tried multiple classification models, such as Random Forest, XGBoost, LGBM, Decision Tree, and AutoML. The Random Forest Classifier, built on data to classify attempts_range, had an accuracy of 61% with an F1_score of 0.605. Similarly, XGBoost had an accuracy of 64% with an F1_score of 0.6373. LGBM was able to provide an accuracy of 64% with a 0.6353 F1_score. Even AutoML was able to provide an accuracy of 64% with a 0.6393 F1_score. Overall, by taking note of the F1_score for each category of attempts as well as over all accuracy and F1_socre(micro), we conclude that XGBoost is the best model for the classification of attempts_range.

5 Summary

In the project, the main objective was to predict attempts range using data from three merged datasets containing user details, problem details, and attempts details. After merging, data cleaning was performed, including missing value imputation and dropping unnecessary columns. A new feature called "days" was created to represent the number of days each user spent on the website. Graphical analysis was conducted to gain insights into the data, followed by statistical testing using parametric and non-parametric tests to assess associations between target and continuous variables. One-way ANOVA and Kruskal-Wallis H test were utilized for this purpose. Additionally, chi-square tests and Cramer V tests were employed to check associations between different categorical variables. The project also involved employing log-linear models to test associations between target variables and other continuous variables. Finally, we proceeded with model building, trying various classifiers like Decision Tree, Random Forest, XGBoost, LGBM, and AutoML, while also experimenting with techniques like upsampling and weight assigning to enhance accuracy. Ultimately, XGBoost was identified as the best model for the current scenario.

Overall, the project aimed to combine and clean diverse datasets to create meaningful features for analysis. Through graphical analysis and statistical tests, the we explored relationships between variables, facilitating data-driven decision-making. The model-building phase involved testing several classifiers and optimization techniques to arrive at the most suitable model for the task at hand. The results achieved in this project provide valuable insights and a foundation for future improvements in the analyzed scenario.

6 Future work

Advanced predictive models like deep learning models can be used to make the model more robust. Also it is better to collect more data to experiment the models and to make the models more generic and robust especially for higher attempts range.

7 Bibliography

Certainly! Below is the bibliography in APA format:

1. Scikit-learn Documentation. https://scikit-learn.org/
2. Keras Documentation. https://keras.io/
3. Towards Data Science https://towardsdatascience.com/
4. Kaggle (Data Science Community). https://www.kaggle.com/
5. Cross Validated (Stats Exchange). https://stats.stackexchange.com/
6. DataCamp (Data Science Learning Platform). https://www.datacamp.com/
7. NIST/SEMATECH e-Handbook of Statistical Methods. https://www.itl.nist.gov/div898/handbook/index.htm
8. Josh Starmer's YouTube Channel. https://www.youtube.com/user/joshstarmer
9. StatisticsHowTo - Chi-Square Test. https://www.statisticshowto.com/chi-square-test/
10. UCLA Statistics Online - Logit Regression. https://stats.idre.ucla.edu/r/dae/logit-regression/
11. Penn State Online Statistics Resources. https://online.stat.psu.edu/stat504/node/168/
12. StatisticsHowTo - Kruskal-Wallis Test. https://www.statisticshowto.com/kruskal-wallis/
13. Penn State Online Statistics Program. https://online.stat.psu.edu/statprogram/r
14. ResearchGate - Scientific Publications. https://www.researchgate.net/
15. Domingos, P. (1997). A few useful things to know about machine learning. http://www.cs.washington.edu/dm/papers/rules_induction.pdf
16. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. https://web.stanford.edu/~hastie/Papers/ESLII.pdf
17. Genewein, T., Foerster, J., Farquhar, G., Whiteson, S., & Rocktäschel, T. (2021). On the Universality of Human-Behavior-Based Stackelberg Equilibria in Sequential Social Dilemmas. https://arxiv.org/abs/2105.04922
18. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. https://arxiv.org/abs/1406.2661

8 Appendix

Python codes:

Importing various libraries required for analysis

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings(’ignore’)
import scipy.stats as stat
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import precision_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import RidgeClassifier
import matplotlib.pyplot as plt
from sklearn.model_selection import RandomizedSearchCV
```

Splitting data to test and train set

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=5,shuffle=True,stratify=
```

Logistic regression

```
model = LogisticRegression(multi_class=’multinomial’)
model.fit(x_trn,y_trn)
```

Function to analyse out come of different ML models

```
import seaborn as sns
import matplotlib.pyplot as plt
def analysis(y_trn, y_train_prdct):
```

```
print(”Classification report”, classification_report(y_trn, y_train_prdct))
print(’accuracy’, accuracy_score(y_trn, y_train_prdct))
print(”precision”, precision_score(y_trn, y_train_prdct, average=”micro”))
print(”recall”, recall_score(y_trn, y_train_prdct, average=”micro”))
cm = confusion_matrix(y_trn, y_train_prdct)
sns.heatmap(cm, annot=True, fmt=”d”, cmap=”Blues”)
plt.xlabel(’Predicted’)
plt.ylabel(’True’)
plt.title(’Confusion Matrix’)
plt.show()
```

Decision Tree

```
#fitting training and testing of Decision Tree
model1=DecisionTreeClassifier() # fitting model
model1.fit(x_trn, y_trn)
#predict
y_test_prdct1 = model1.predict(x_tst)
y_train_prdct1 = model1.predict(x_trn)
print(”for train”)
analysis(y_trn, y_train_prdct1)
print(”for test”)
analysis(y_tst, y_test_prdct1)
```

Random Forest Classifier

```
#fitting training and testing of RandomForest CLassifier
#define model
model2 = RandomForestClassifier()
model2.fit(x_trn, y_trn)
y_test_prdct2 = model2.predict(x_tst)
y_train_prdct2 = model2.predict(x_trn)
#analysis of model
print(”Train”)
analysis(y_trn, y_train_prdct2)
print(”Test”)
analysis(y_tst, y_test_prdct2)
```

XGBoost

```
#fitting training and testing of XGBoost
from xgboost import XGBClassifier
modelxg = XGBClassifier()
```

```
modelxg.fit(x_train,y_train)
y_train_pred = modelxg.predict(x_trn)
y_test_pred = modelxg.predict(x_tst)
print(”Train”)
analysis(y_trn, y_train_pred)
print(”Test”)
analysis(y_tst, y_test_pred)
```

LGBM ;

```
#fitting training and testing of LGBM
from lightgbm import LGBMClassifier
modellgbm = LGBMClassifier(learning_rate=0.09, max_depth=-5, random_state=42)
modellgbm.fit(x_trn, y_trn)
y_test_predct = modellgbm.predict(x_tst)
y_train_predct = modellgbm.predict(x_trn)
print(”Train acrcy”, f1_score(y_train_predct, y_trn, average=’micro’).round(4))
print(”Test acrcy”, f1_score(y_test_predct, y_tst, average=’micro’).round(4))
print(”Train”)
analysis(y_trn, y_train_predct)
print(”Test”)
analysis(y_tst, y_test_predct)
```

Auto ML:

```
#fitting training and testing of AutoML
from lightgbm import LGBMClassifier
modellgbm = LGBMClassifier(learning_rate=0.09, max_depth=-5, random_state=42)
modellgbm.fit(x_trn, y_trn)
y_test_predct = modellgbm.predict(x_tst)
y_train_predct = modellgbm.predict(x_trn)


print(”Train acrcy”, f1_score(y_train_predct, y_trn, average=’micro’).round(4))
print(”Test acrcy”, f1_score(y_test_predct, y_tst, average=’micro’).round(4))
print(”Train”)
analysis(y_trn, y_train_predct)
print(”Test”)
analysis(y_tst, y_test_predct)
```