## horizontal line



**AMRITA SCHOOL OF ENGINEERING**

**19CSE304 - Foundations of Data Science**

CASE STUDY REVIEW TOPIC:

Problem Recommendation

**STUDENT DETAILS:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **NAME** | **ROLL NO** |
| 1 | Guhan M | CB.EN.U4CSE19425 |
| 2 | M Sri Hari | CB.EN.U4CSE19435 |
| 3 | Praveen Kumar R | CB.EN.U4CSE19451 |
| 4 | S Pranav Adith | CB.EN.U4CSE19458 |

# Overview

The use of coding platforms to support students acquiring programming skills is common nowadays because this type of software contains a large collection of programming exercises to be solved by students.

A common problem that students face when using coding platforms is information overload, as choosing the right problem to solve can be quite frustrating due to the large number of problems offered. Hence, the aim of this paper is to support students with the information overload problem by using a collaborative filtering recommendation approach that filters out programming problems suitable for students' programming skills.

It uses an enriched user-problem matrix that implies a better student role representation, facilitating the computation of closer neighborhoods and hence a more accurate recommendation.

A case study is carried out on a coding platform real dataset showing that the proposal outperforms other previous approaches.

# Goals

In this challenge we are required to build a model to predict the number of attempts taken by participants for a successful submission to online programming challenges. Data of programmers and questions they solved previously were given along with the time they took to solve the questions.

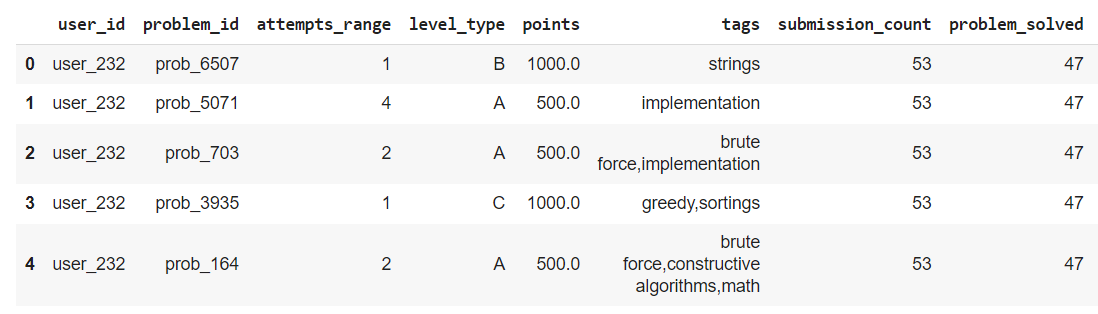
# 

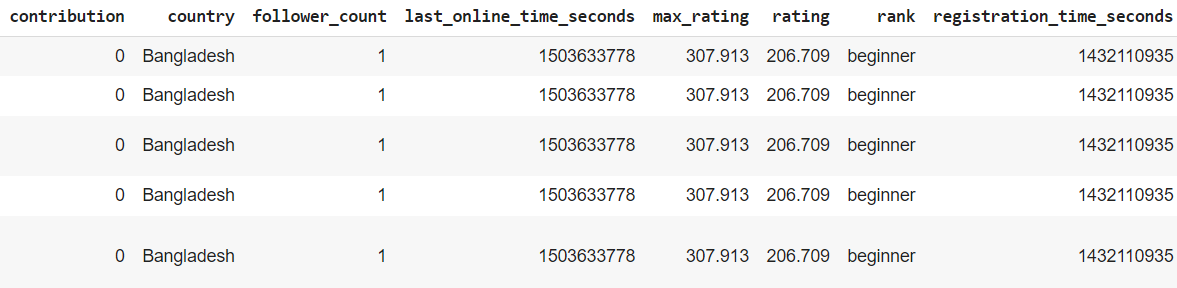
# About dataset

In order to analyzing users we have used

Attributes details:

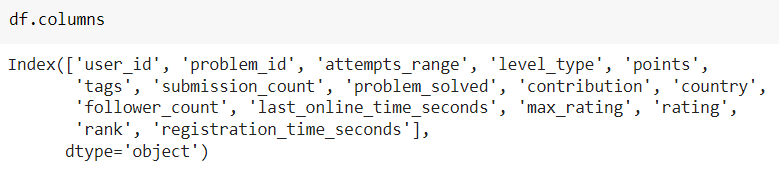
* **user\_id**
* **problem\_id**
* **attempts\_range**
* **level\_type points**
* **tags**
* **submission\_count**
* **problem\_solved**
* **contribution**
* **country**
* **follower\_count**
* **last\_online\_time\_seconds**
* **max\_rating**
* **rating**
* **rank**
* **Registration\_time\_seconds**

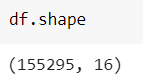
****

****

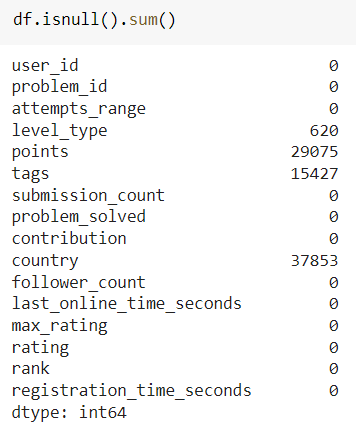
# Data Preprocessing

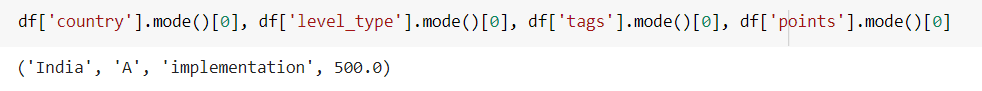
The first step in the process of analyzing air quality is to pre-process the dataset obtained. It is a way of converting this raw data into a much-desired form so that useful information can be derived from it, which is fed into the training model. Our dataset has few null values and outliers which must be handled using necessary methods.

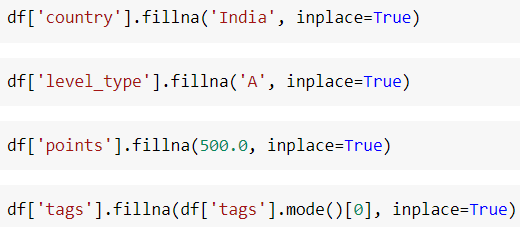




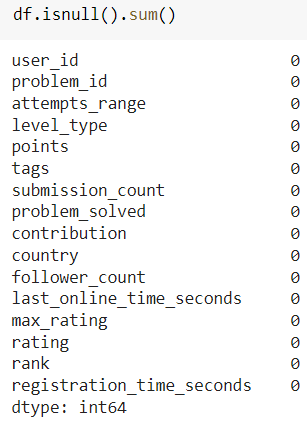
**CHECKING FOR NAN VALUES**







**HERE NAN VALUES WERE REMOVED**



# Outlier Detection

# Outliers increase the variability in your data, which decreases statistical power

# 

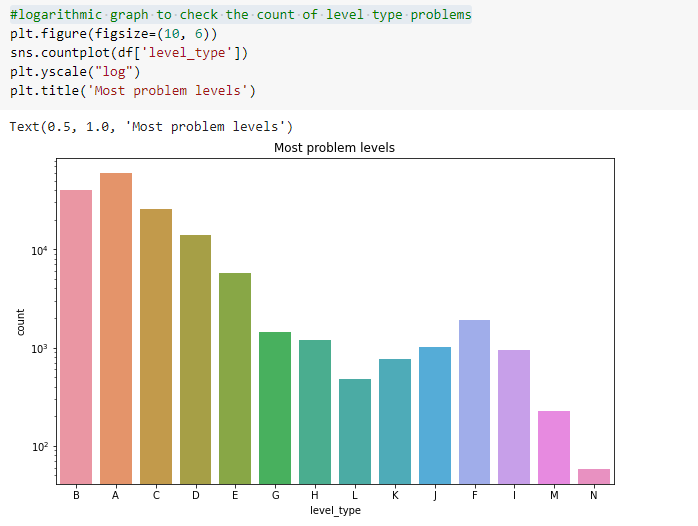
# 

# 

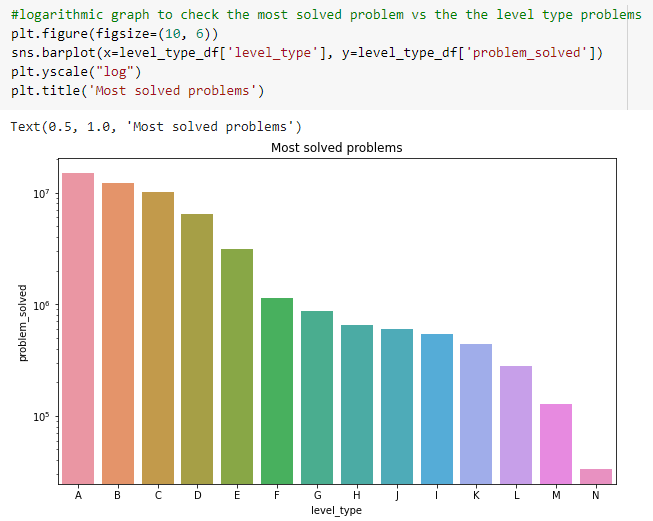
# Visualization

For easy understanding we use visualization for graphical representation

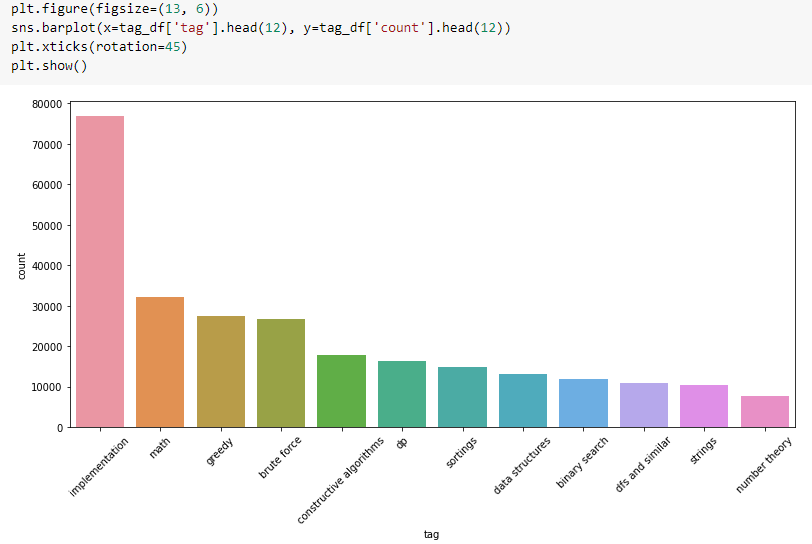
**logarithmic graph to check the count of level type problems**



**logarithmic graph to check the most solved problem vs the level type problems**

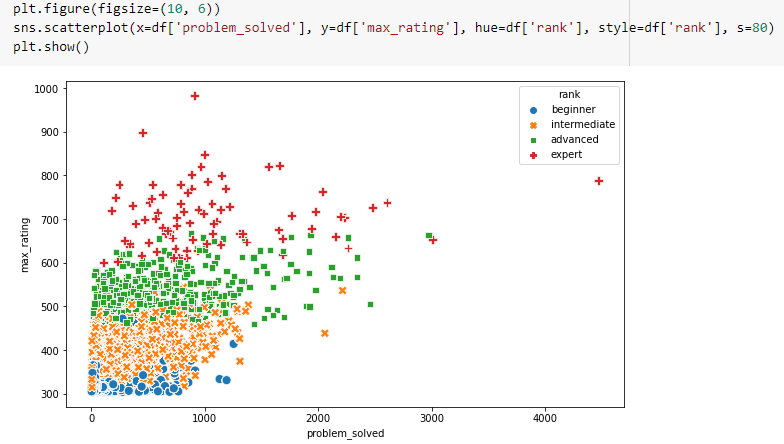


**This graph to check the count of tag**

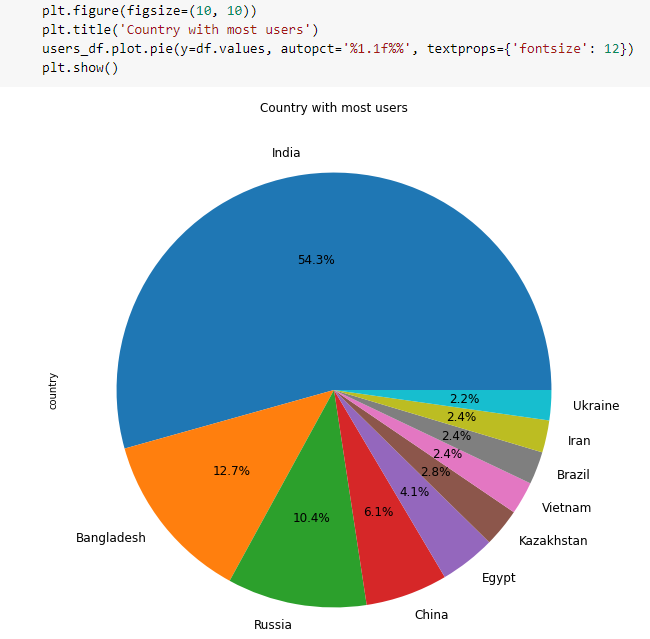


***inference:*** implementation has higher counts than any other tags

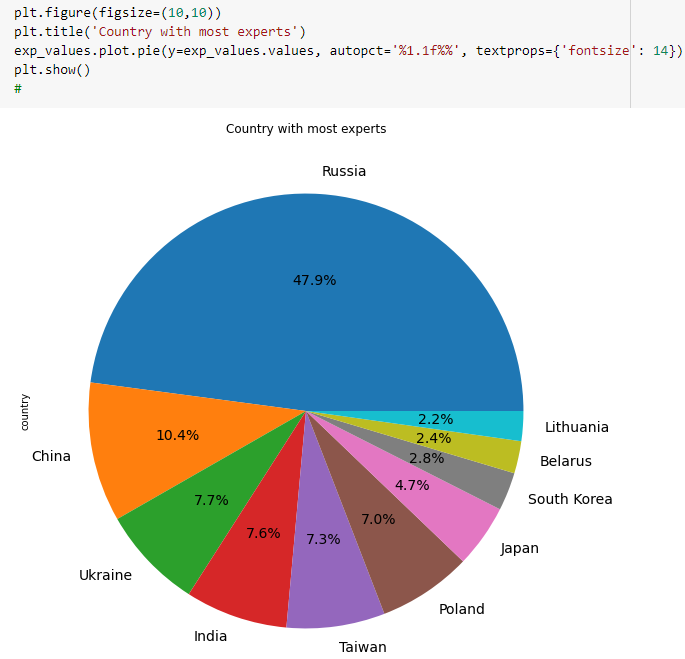
**This scatterplot shows us solved problems vs maximum rating with hue: rank**



***inference:*** experts data are spread unevenly and their max ratings are higher

**Here we can see the users from different countries** 

***inference:*** India has the most no. of users

**Here we can see the experts users from different countries** 

***inference:*** Russia has the most no. of experts

# 

# 

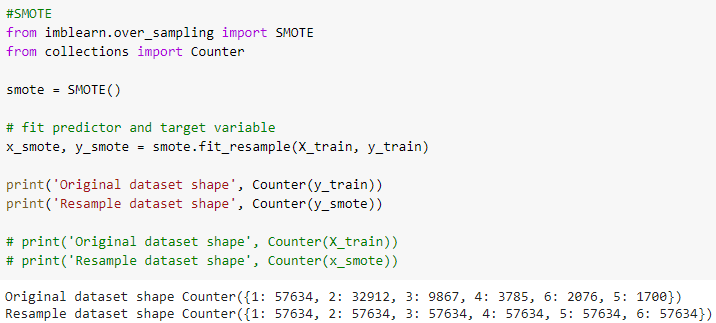
# Handling Imbalance Data

Imbalanced datasets mean that the number of observations differs for the classes in a classification dataset. This imbalance can lead to inaccurate and biased results while building the model. The distribution of an imbalanced dataset is characterized by very high differences between the classes involved. To handle this, techniques such as oversampling and under sampling have been used.

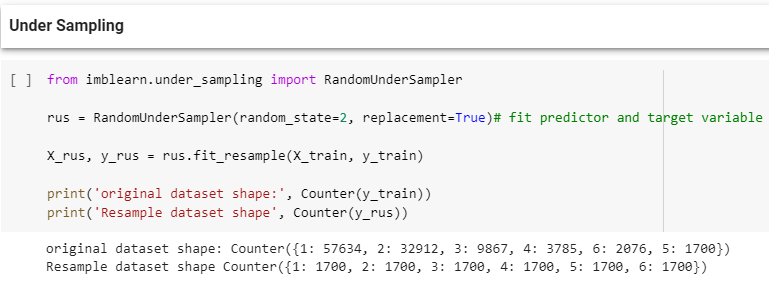
CHECKING IMBALANCE OR NOT:

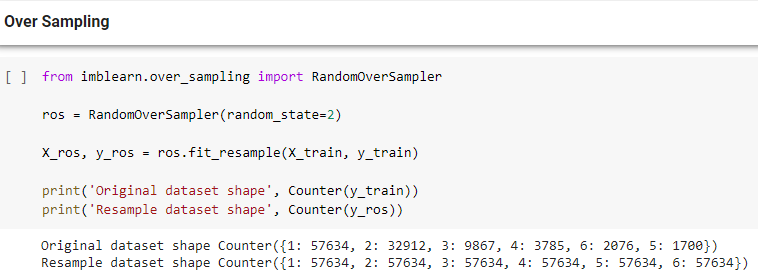
SMOTE[Synthetic Minority Oversampling Technique.]

This is a statistical technique for increasing the number of cases in your dataset in a balanced way. The module works by generating new instances from existing minority cases that you supply as input.



**Under and Over sampling technique:**

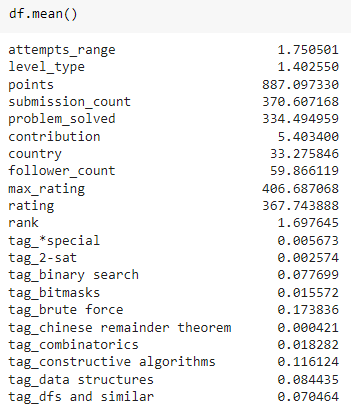
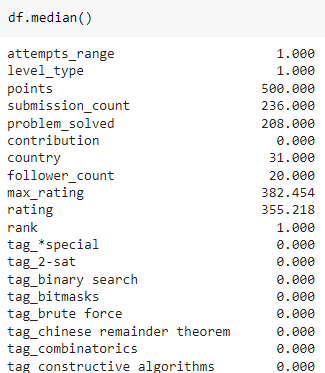
After splitting the data into TRAIN and TEST data we have to apply the technique



# Descriptive Statistics

There are 3 main types of descriptive statistics:

* The distribution concerns the frequency of each value.
* The central tendency concerns the averages of the values.
* The variability or dispersion concerns how spread out the values are

# 

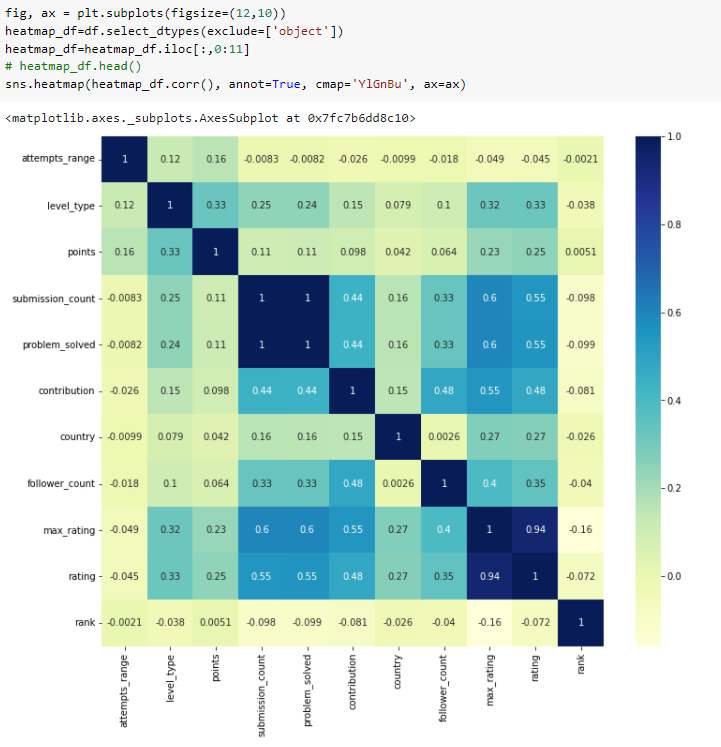
# 

# 

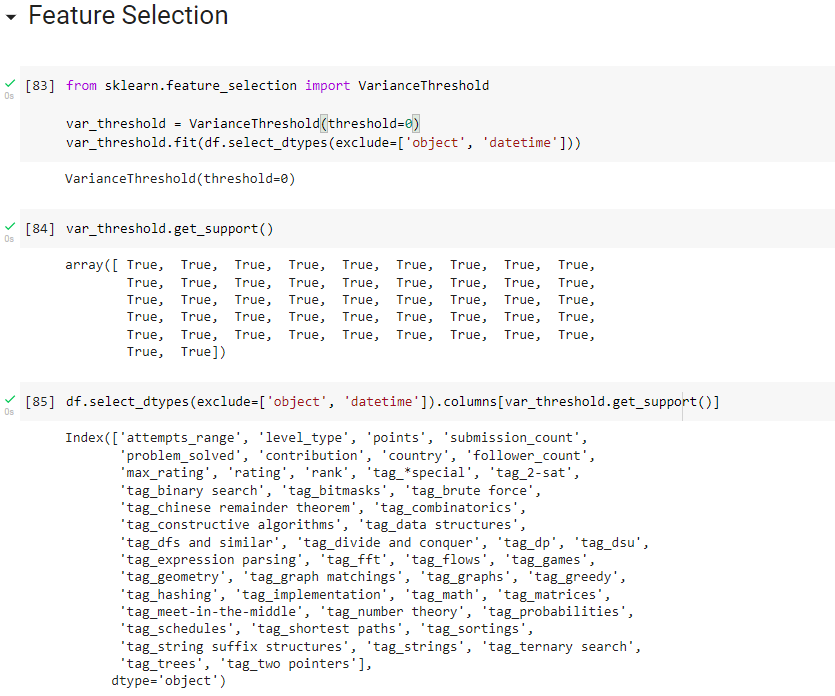
# 

# 

# Removing duplicated and highly correlated columns



# Feature Selection



# 

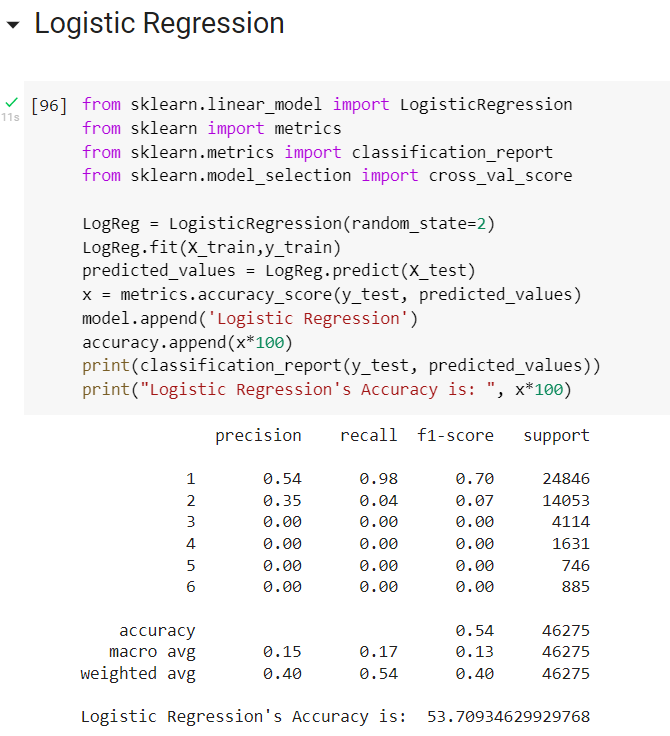
# 

# 

# Model Building

Logistic Regression

It establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit straight line



**The accuracy for X\_train and y\_train is 53.7%. The precisions for 3, 4, 5, 6 are very low because of imbalanced data**

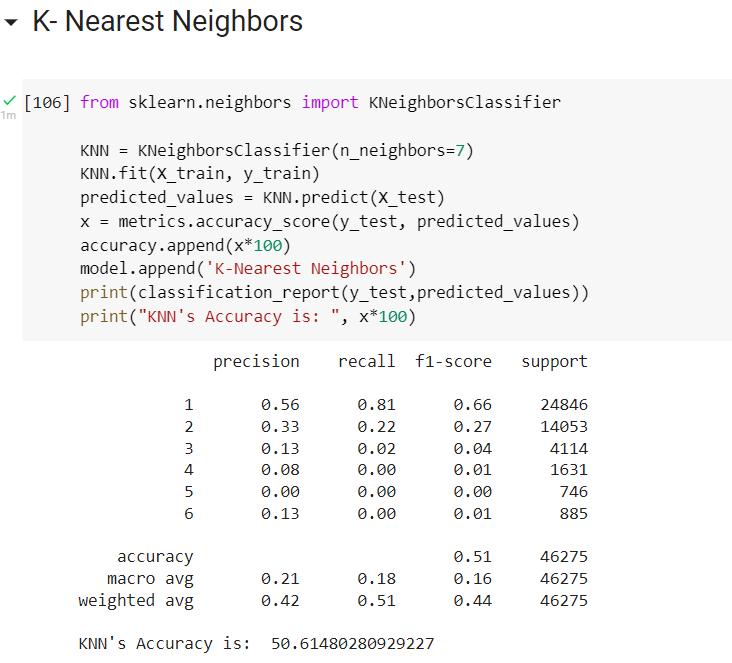
Random Forest Classifier

Random forest classifiers can be used to solve regression or classification problems. The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble consists of a data sample drawn from a training set with replacement, called the bootstrap sample.



K-Nearest Neighbors

K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of ‘K’ training data and which class holds the highest probability will be selected. In the case of regression, the value is the mean of the ‘K’ selected training points.



# Model Comparison

# 

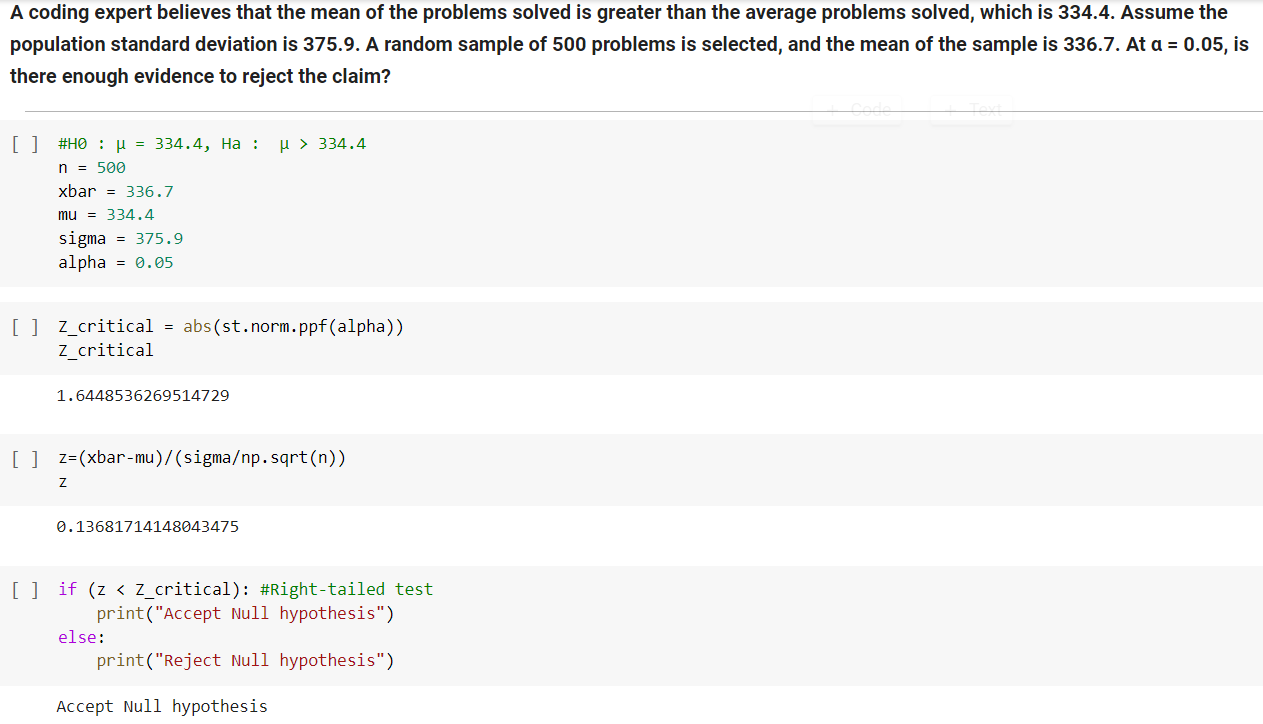
# 

# 

# Hypothesis Testing

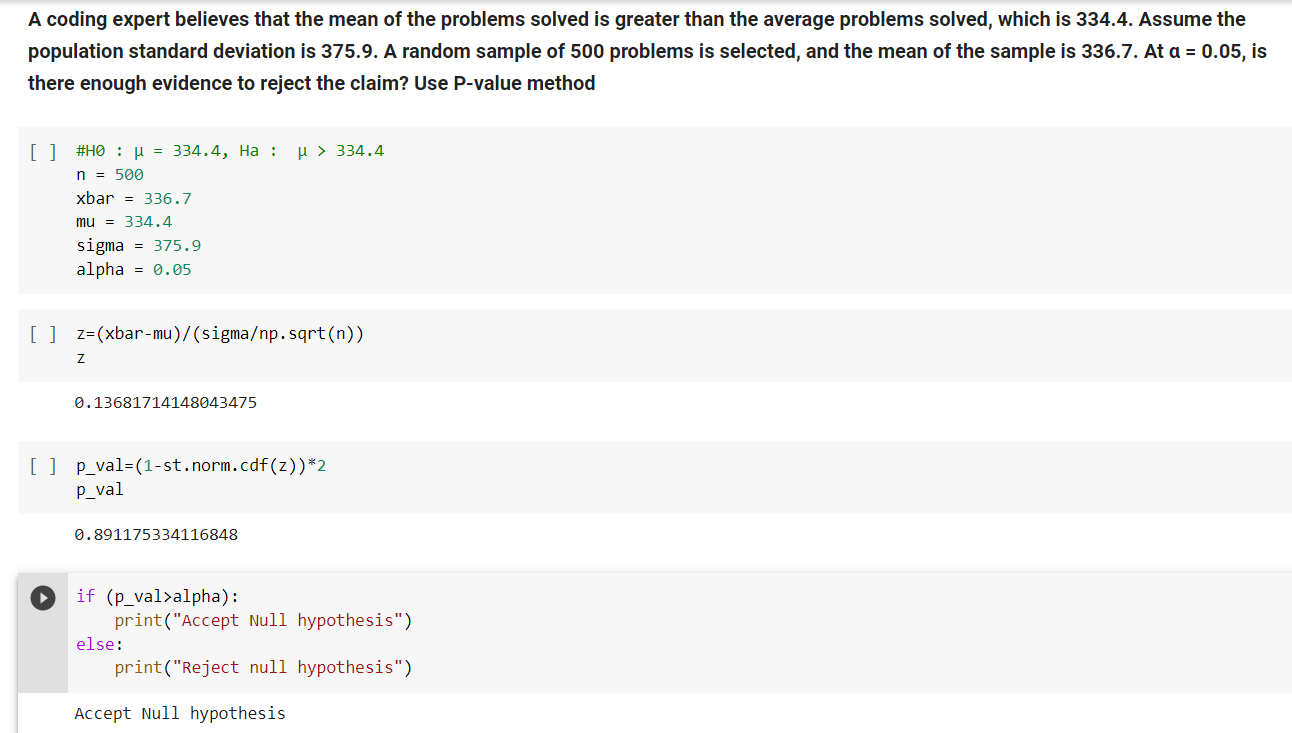
**Z-test**

**A coding expert believes that the mean of the problems solved is greater than the average problems solved, which is 334.4. Assume the population standard deviation is 375.9. A random sample of 500 problems is selected, and the mean of the sample is 336.7. At α = 0.05, is there enough evidence to reject the claim?**

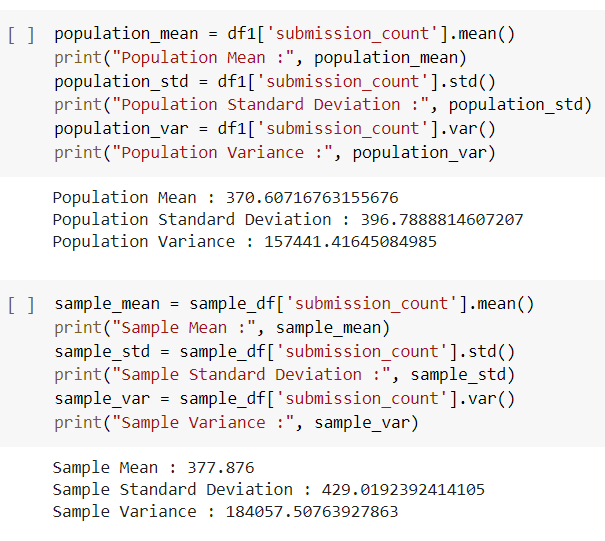


**Z-Test using P-value**

**A coding expert believes that the mean of the problems solved is greater than the average problems solved, which is 334.4. Assume the population standard deviation is 375.9. A random sample of 500 problems is selected, and the mean of the sample is 336.7. At α = 0.05, is there enough evidence to reject the claim? Use P-value method**



**t-Test**

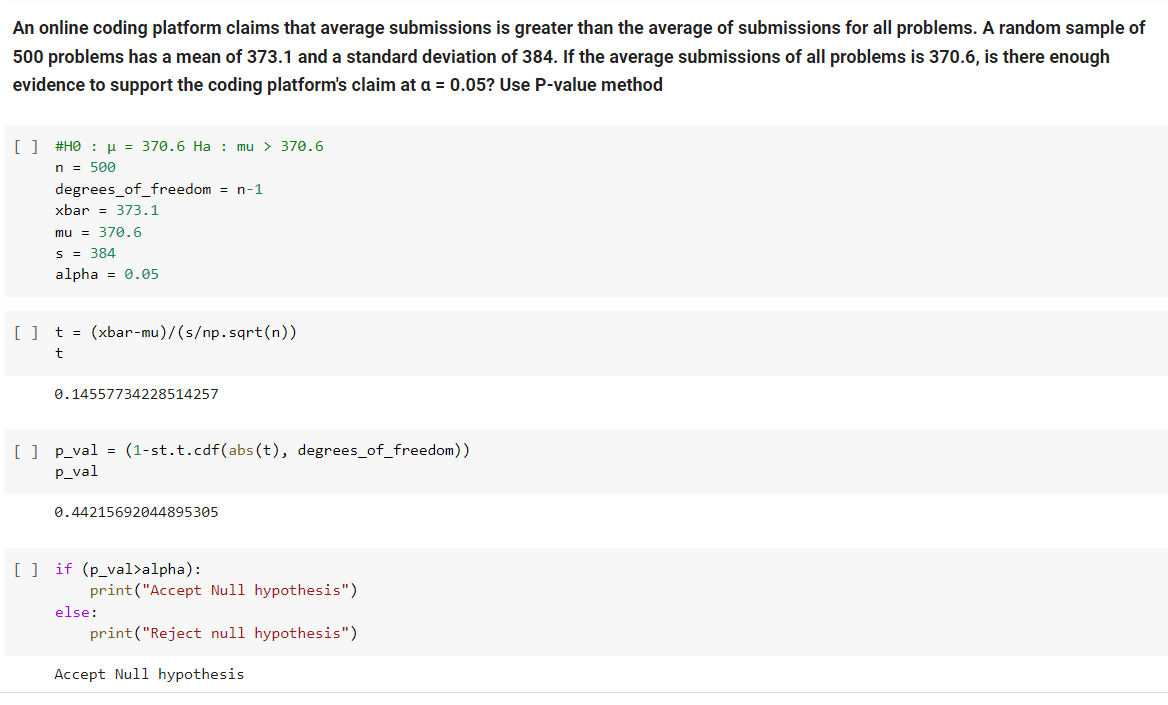
****

**An online coding platform claims that the average submissions is 370.6. A random sample of 500 problems had a mean submission of 373.1. The sample standard deviation is 384.0. Is there enough evidence to reject the coding platform’s claim at α = 0.05? Assume the variable is normally distributed.**

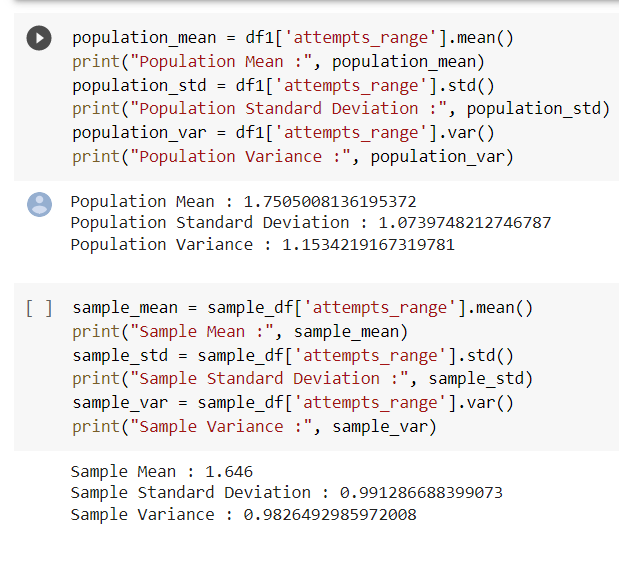
# 

**t-Test using P-value**

**An online coding platform claims that average submissions is greater than the average of submissions for all problems. A random sample of 500 problems has a mean of 373.1 and a standard deviation of 384. If the average submissions of all problems is 370.6, is there enough evidence to support the coding platform's claim at α = 0.05? Use P-value method**

****

**Chi-Square Test**

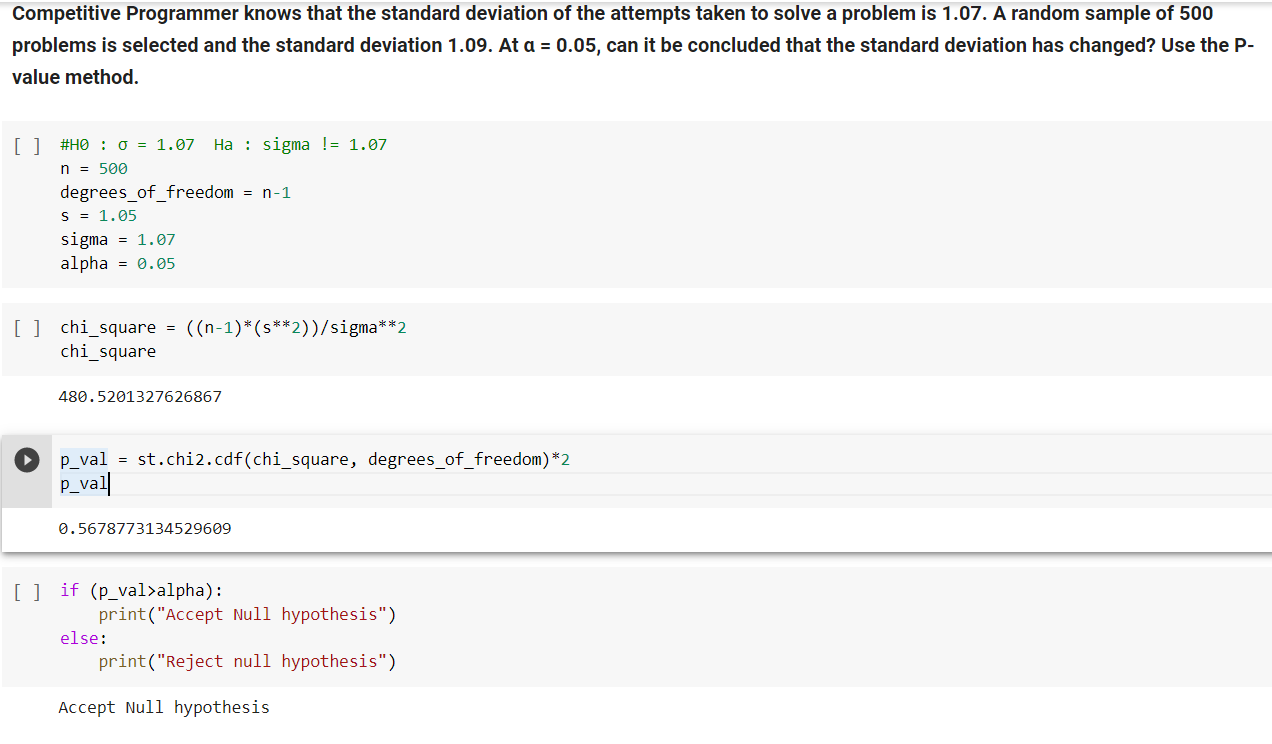
****

**Competitive Programmer wishes to see if the variance of attempts taken of 500 problems is less than the variance of the population, which is 1.15. The variance of the attempts taken of 500 problems was 1.2. Test the claim at α = 0.05**

****

**Chi-Square Test using P-Value**

**A Competitive Programmer knows that the standard deviation of the attempts taken to solve a problem is 1.07. A random sample of 500 problems is selected and the standard deviation 1.09. At α = 0.05, can it be concluded that the standard deviation has changed? Use the P-value method.**

****

# Conclusion

The selected dataset has been preprocessed by removing irrelevant columns, replacement of null values with empty spaces and reformation of strings without special/Numeric characters. Appropriate columns have been selected and visualized for getting insightful inference regarding the dataset using seaborn and matplotlib packages. The main objective of building a better model to analyze the recommended problems for the competitive coding program….!!!

# 

