Recommendation Engine

Online Programming Platform

Team Members:

Dhaval Suthar

Prajakta Sumbe

Ravi Kiran

Veena Iyer

Date: 04/27/2019

Overview & Purpose

Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications. The aim of competitive programming is to write the source code of computer programs which are able to solve given problems. Major companies hire from top coding platforms. The planning, workforce, time and money which goes into recruiting is cut to half by the competitive coding platforms.

Docker - https://hub.docker.com/r/suthardhaval24/ds_finalproject

Heroku -

Git Hub - https://github.com/DS2019Spring/Final_Repository

GOALS

1. Recommending the questions that a programmer should solve given his/her current expertise is a big challenge for Online Programming Platforms but is an essential task to judge a programmer's expertise in that particular area which will help companies in their hiring process.

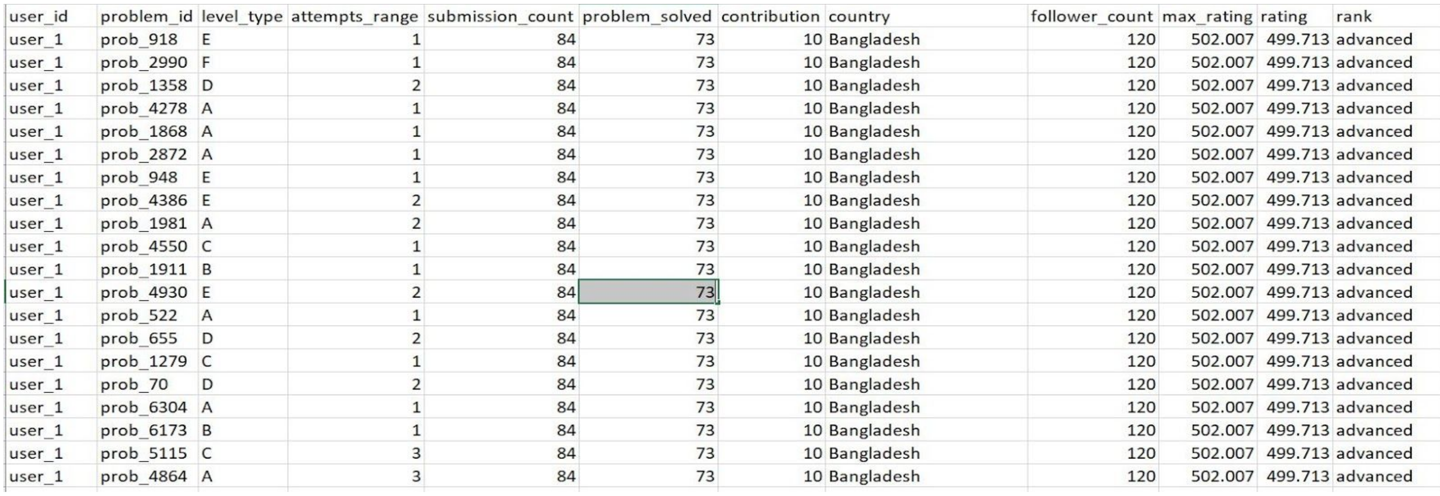
USE CASES

1. Student - Can be used to provide the suggested questions to the user based on user profile.

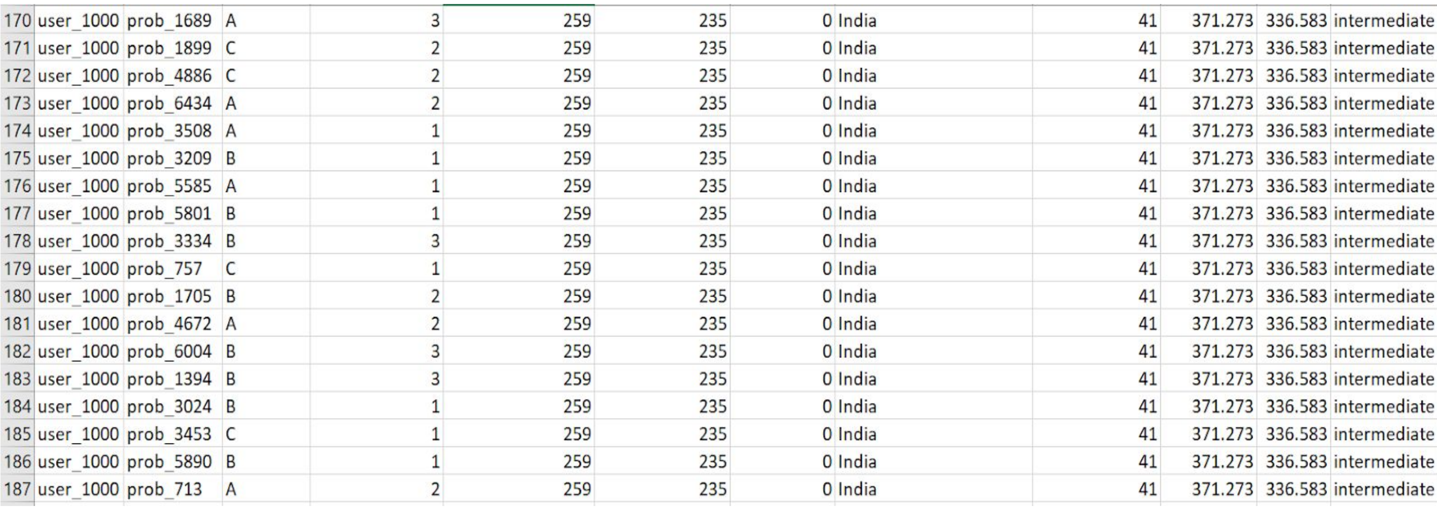
2. Online Platforms/Company: Can use the predicted attempts to evaluate the expertise of the user and suggest questions appropriately.

Data Ingestion

We will work with the data wherein the features are as below:



user_id	problem_id	level_type	attempts_range	submission_count	problem_solved	contribution	country	follower_count	max_rating	rating	rank
user_1	prob_918	E	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_2990	F	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_1358	D	2	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_4278	A	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_1868	A	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_2872	A	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_948	E	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_4386	E	2	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_1981	A	2	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_4550	C	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_1911	B	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_4930	E	2	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_522	A	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_655	D	2	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_1279	C	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_70	D	2	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_6304	A	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_6173	B	1	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_5115	C	3	84	73	10	Bangladesh	120	502.007	499.713	advanced
user_1	prob_4864	A	3	84	73	10	Bangladesh	120	502.007	499.713	advanced



170	user_1000	prob_1689	A	3	259	235	0	India	41	371.273	336.583	intermediate
171	user_1000	prob_1899	C	2	259	235	0	India	41	371.273	336.583	intermediate
172	user_1000	prob_4886	C	2	259	235	0	India	41	371.273	336.583	intermediate
173	user_1000	prob_6434	A	2	259	235	0	India	41	371.273	336.583	intermediate
174	user_1000	prob_3508	A	1	259	235	0	India	41	371.273	336.583	intermediate
175	user_1000	prob_3209	B	1	259	235	0	India	41	371.273	336.583	intermediate
176	user_1000	prob_5585	A	1	259	235	0	India	41	371.273	336.583	intermediate
177	user_1000	prob_5801	B	1	259	235	0	India	41	371.273	336.583	intermediate
178	user_1000	prob_3334	B	3	259	235	0	India	41	371.273	336.583	intermediate
179	user_1000	prob_757	C	1	259	235	0	India	41	371.273	336.583	intermediate
180	user_1000	prob_1705	B	2	259	235	0	India	41	371.273	336.583	intermediate
181	user_1000	prob_4672	A	2	259	235	0	India	41	371.273	336.583	intermediate
182	user_1000	prob_6004	B	3	259	235	0	India	41	371.273	336.583	intermediate
183	user_1000	prob_1394	B	3	259	235	0	India	41	371.273	336.583	intermediate
184	user_1000	prob_3024	B	1	259	235	0	India	41	371.273	336.583	intermediate
185	user_1000	prob_3453	C	1	259	235	0	India	41	371.273	336.583	intermediate
186	user_1000	prob_5890	B	1	259	235	0	India	41	371.273	336.583	intermediate
187	user_1000	prob_713	A	2	259	235	0	India	41	371.273	336.583	intermediate

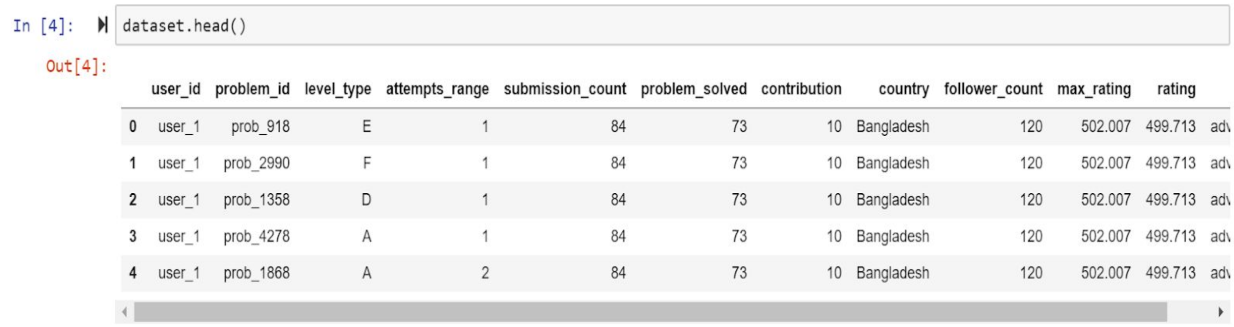
PROCESS OUTLINE

1. Data Preprocessing
   - ➔ Preparing final dataset by joining three subsets of data
   - ➔ Data Cleaning, handling missing values.
2. Exploratory Data Analysis.
3. Study supervised approaches and select the best model for prediction.
4. Design a pipeline and system to implement this approach.
5. Deploy the model.

Exploratory Data Analysis

For exploring the various facets of our data we performed the following exploratory data analysis

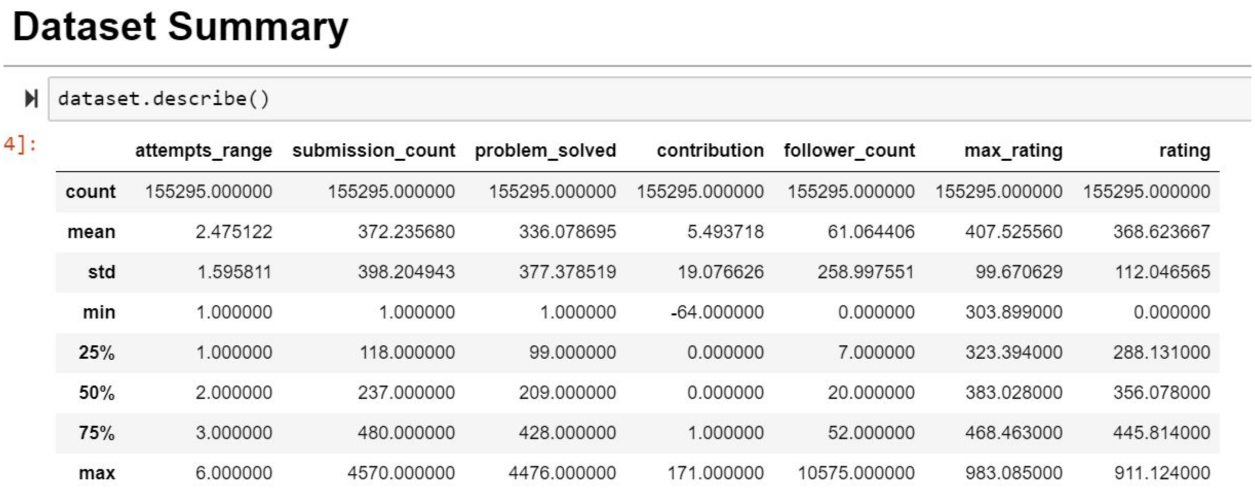
The data head consists of -



```
In [4]:  dataset.head()
Out[4]:
```

	user_id	problem_id	level_type	attempts_range	submission_count	problem_solved	contribution	country	follower_count	max_rating	rating	
0	user_1	prob_918	E	1	84	73	10	Bangladesh	120	502.007	499.713	adv
1	user_1	prob_2990	F	1	84	73	10	Bangladesh	120	502.007	499.713	adv
2	user_1	prob_1358	D	1	84	73	10	Bangladesh	120	502.007	499.713	adv
3	user_1	prob_4278	A	1	84	73	10	Bangladesh	120	502.007	499.713	adv
4	user_1	prob_1868	A	2	84	73	10	Bangladesh	120	502.007	499.713	adv

To summarize the description of the dataset

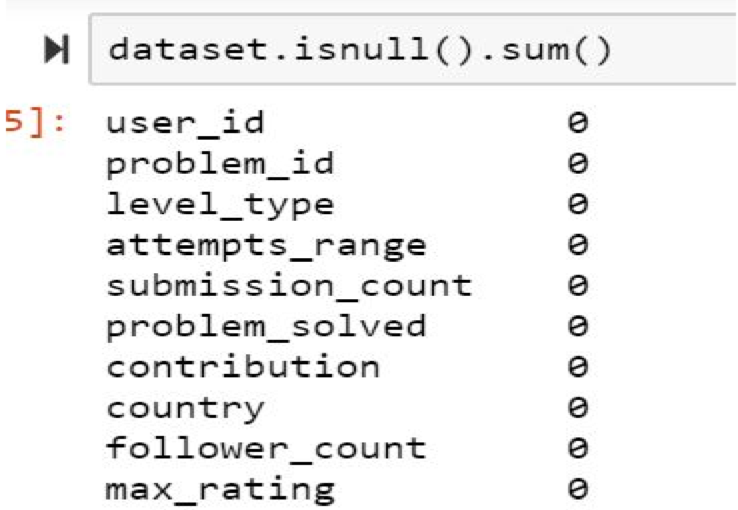


Dataset Summary

```
dataset.describe()
4]:
```

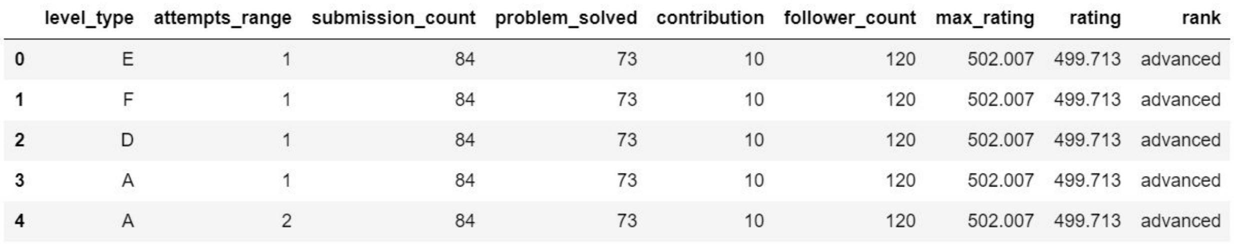
	attempts_range	submission_count	problem_solved	contribution	follower_count	max_rating	rating
count	155295.000000	155295.000000	155295.000000	155295.000000	155295.000000	155295.000000	155295.000000
mean	2.475122	372.235680	336.078695	5.493718	61.064406	407.525560	368.623667
std	1.595811	398.204943	377.378519	19.076626	258.997551	99.670629	112.046565
min	1.000000	1.000000	1.000000	-64.000000	0.000000	303.899000	0.000000
25%	1.000000	118.000000	99.000000	0.000000	7.000000	323.394000	288.131000
50%	2.000000	237.000000	209.000000	0.000000	20.000000	383.028000	356.078000
75%	3.000000	480.000000	428.000000	1.000000	52.000000	468.463000	445.814000
max	6.000000	4570.000000	4476.000000	171.000000	10575.000000	983.085000	911.124000

We then performed data cleaning by checking and removing null values if any



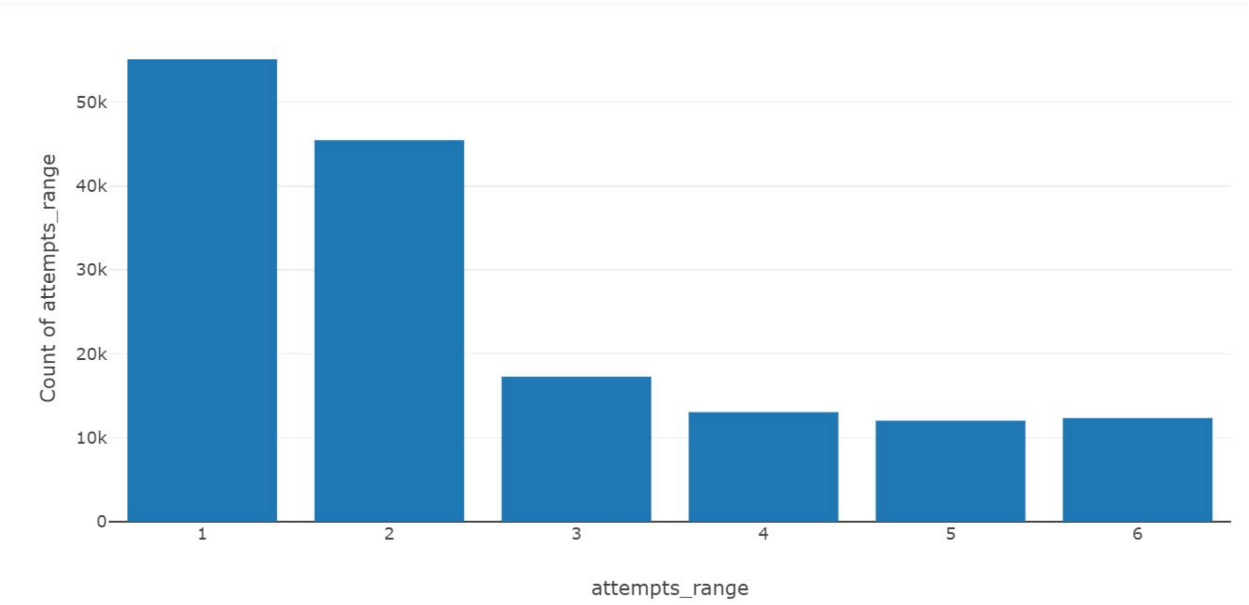
```
dataset.isnull().sum()

5]:  user_id             0
     problem_id          0
     level_type          0
     attempts_range      0
     submission_count    0
     problem_solved      0
     contribution        0
     country             0
     follower_count      0
     max_rating          0
```

Feature selection then comes into picture wherein few features are weighed against each other and appropriate feature is selected.

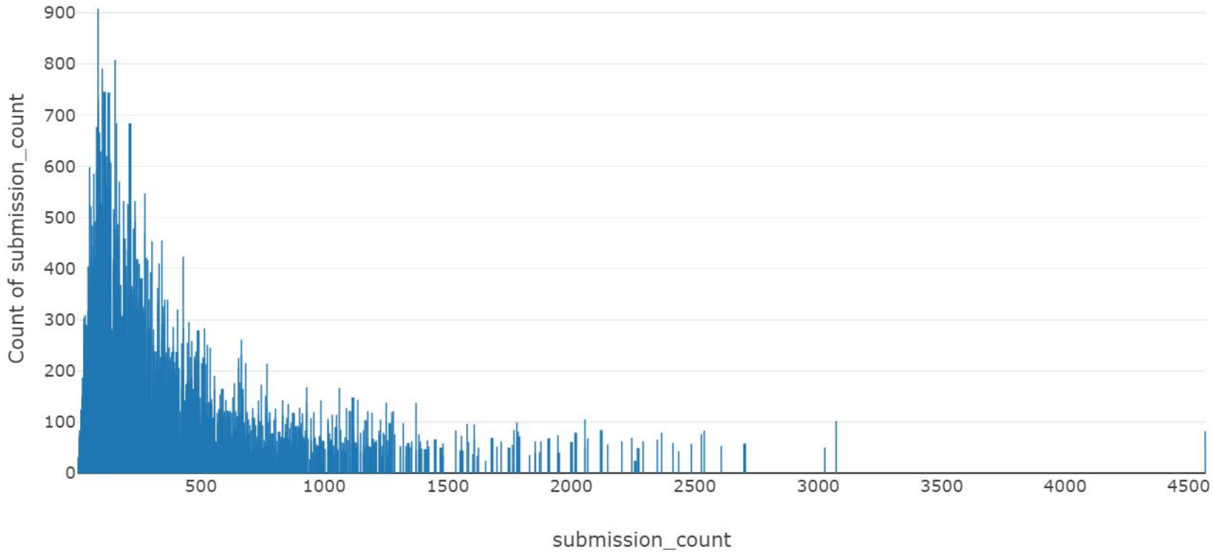


	level_type	attempts_range	submission_count	problem_solved	contribution	follower_count	max_rating	rating	rank
0	E	1	84	73	10	120	502.007	499.713	advanced
1	F	1	84	73	10	120	502.007	499.713	advanced
2	D	1	84	73	10	120	502.007	499.713	advanced
3	A	1	84	73	10	120	502.007	499.713	advanced
4	A	2	84	73	10	120	502.007	499.713	advanced

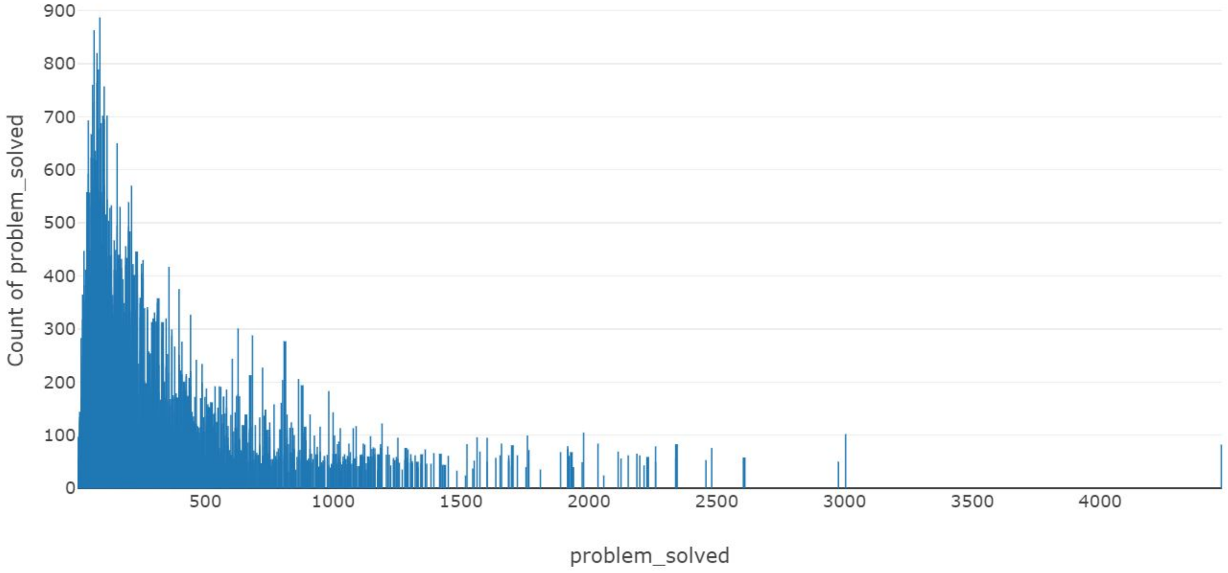
The data has an attempt range that ranges from 1 to 6 which denotes the number of attempts made to solve a question. The count of the attempts throughout the data is as below.



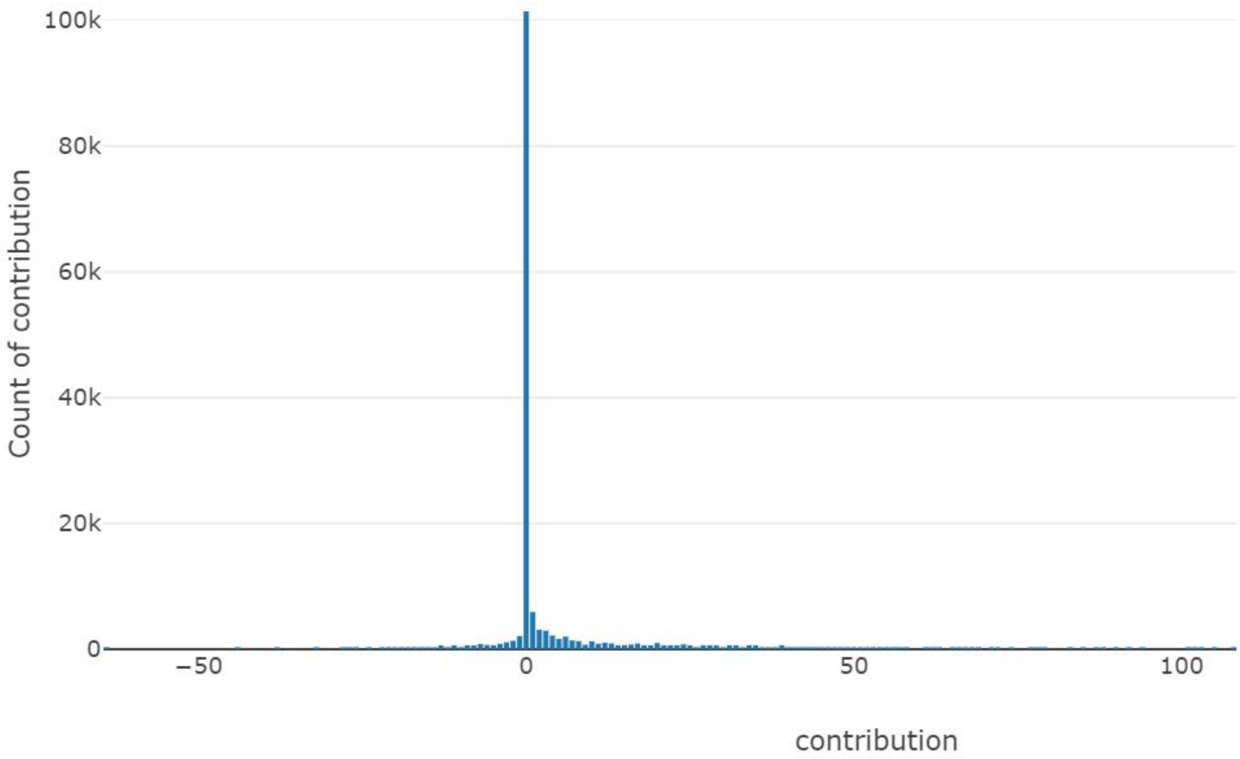
The submission count indicates the number of submissions done by the user



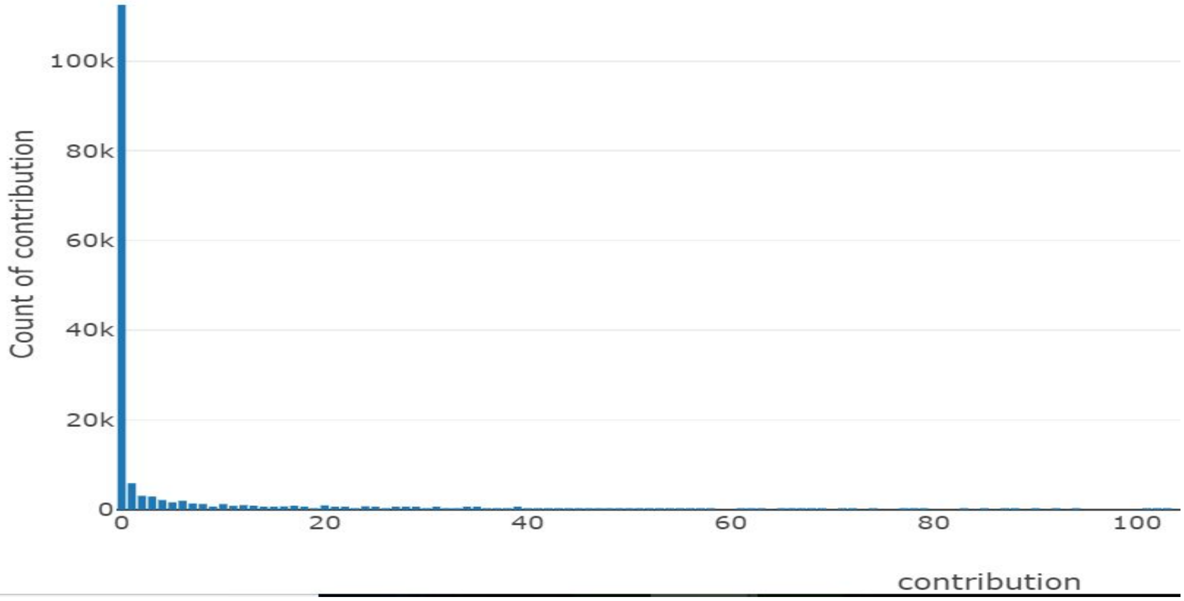
The problem solved indicates the number of problems solved.



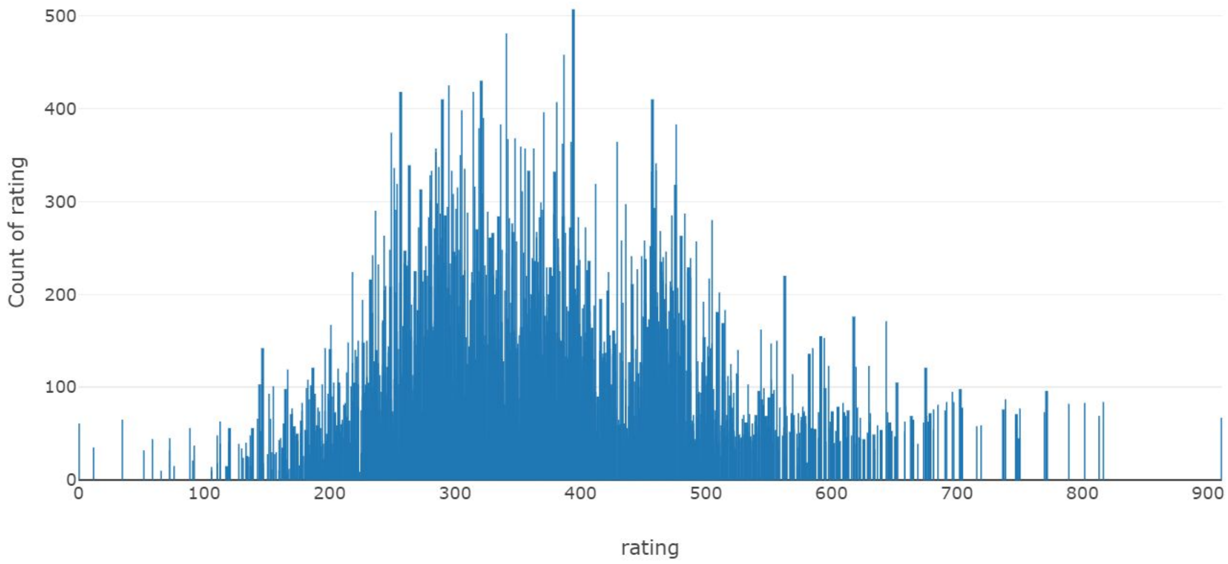
Contribution indicates what the user would contribute to online programming dynamics stated below.



From the above graph we can see that contribution has negative data and thus to clean the data and to regraph contribution.

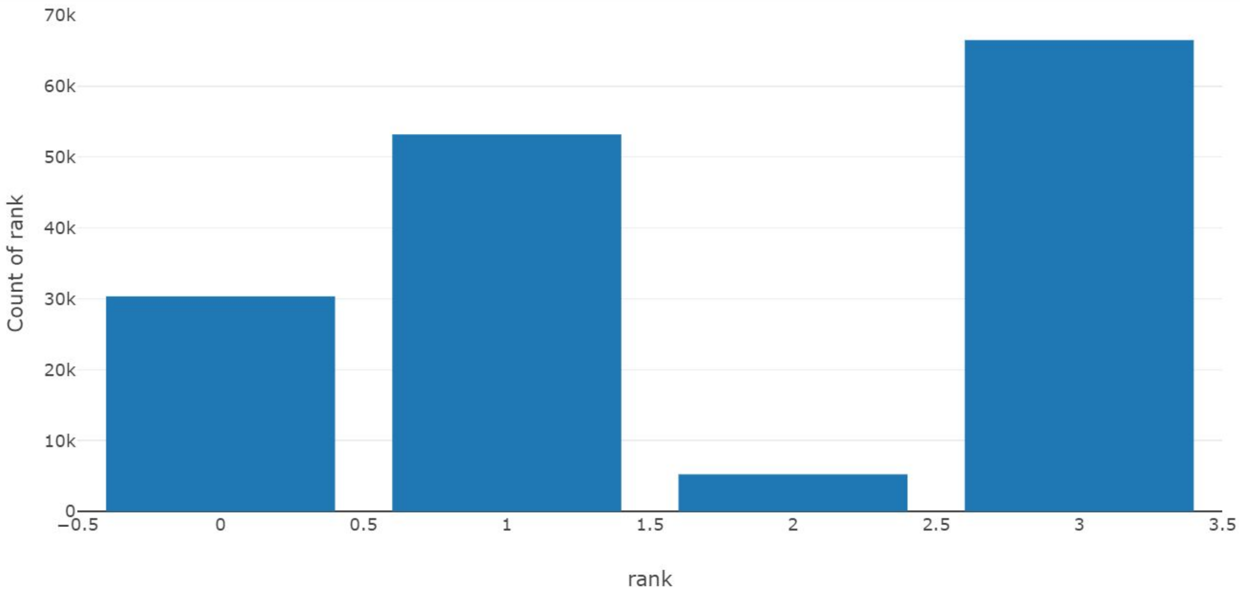


Rating is provided to user as per the above features mentioned judging the overall performance of a user.

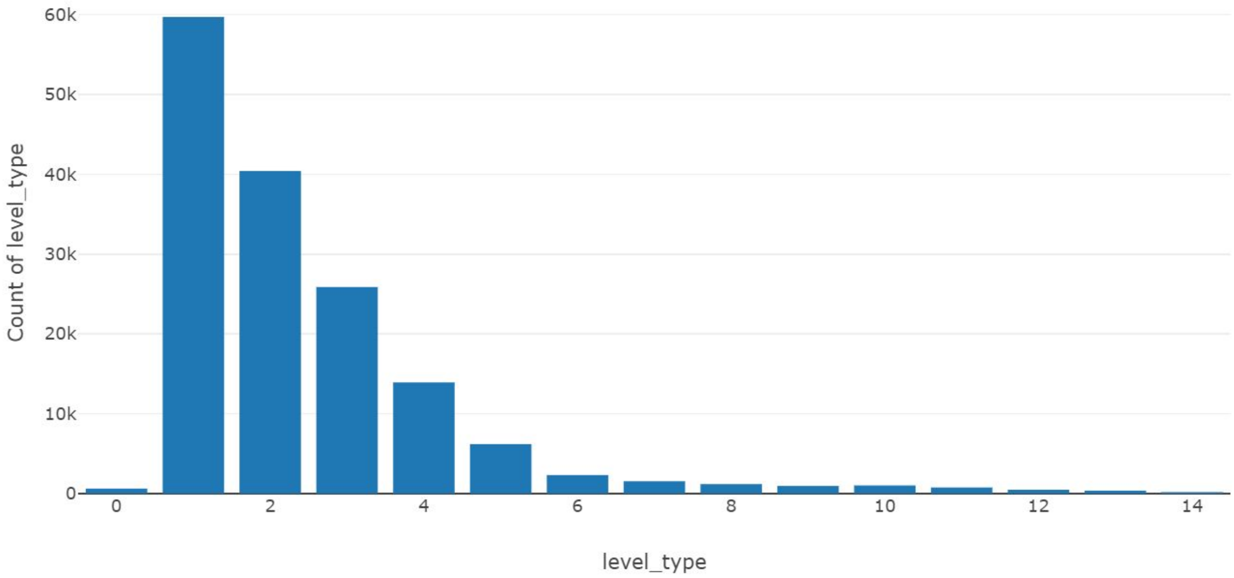


Count of ranks involve four levels which is encoded with label encoder -

Beginner, Intermediate, Advanced, Expert

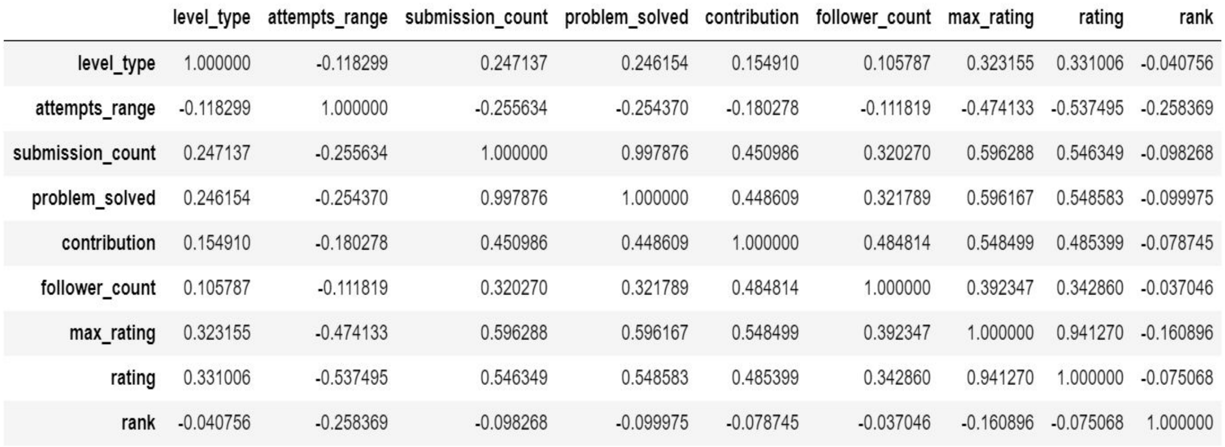


Level type indicates the complexity of the problem which is also labelled using the label encoder.

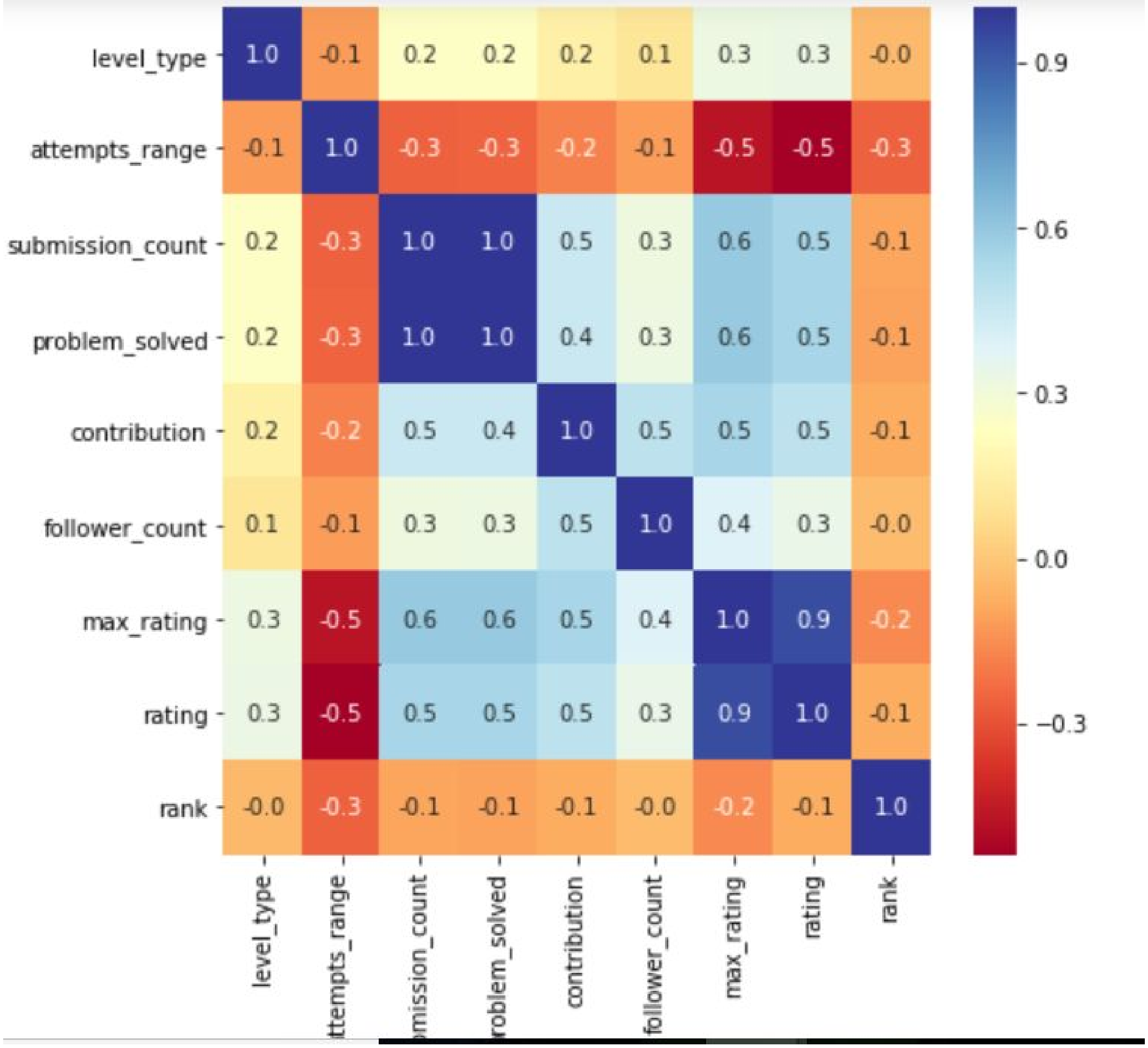


Correlation matrix explains the relation or the correlation between all the other features.

The numeric correlation data and its plots for our dataset is as below -

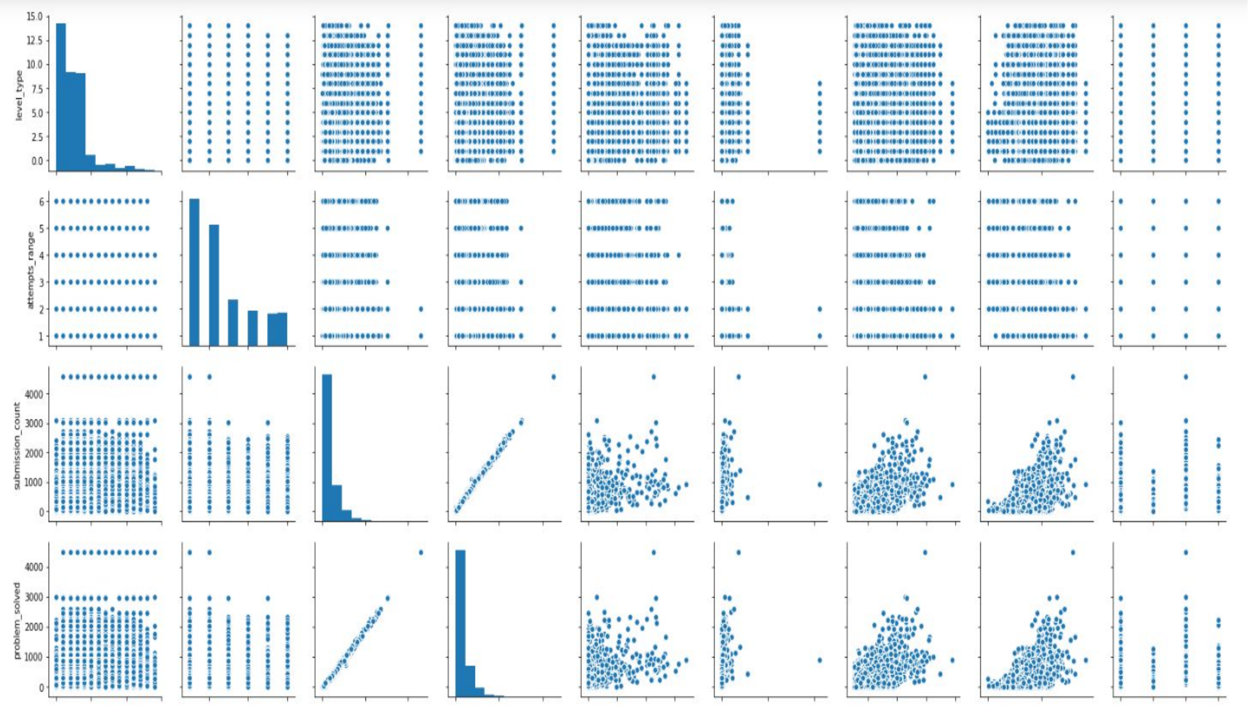


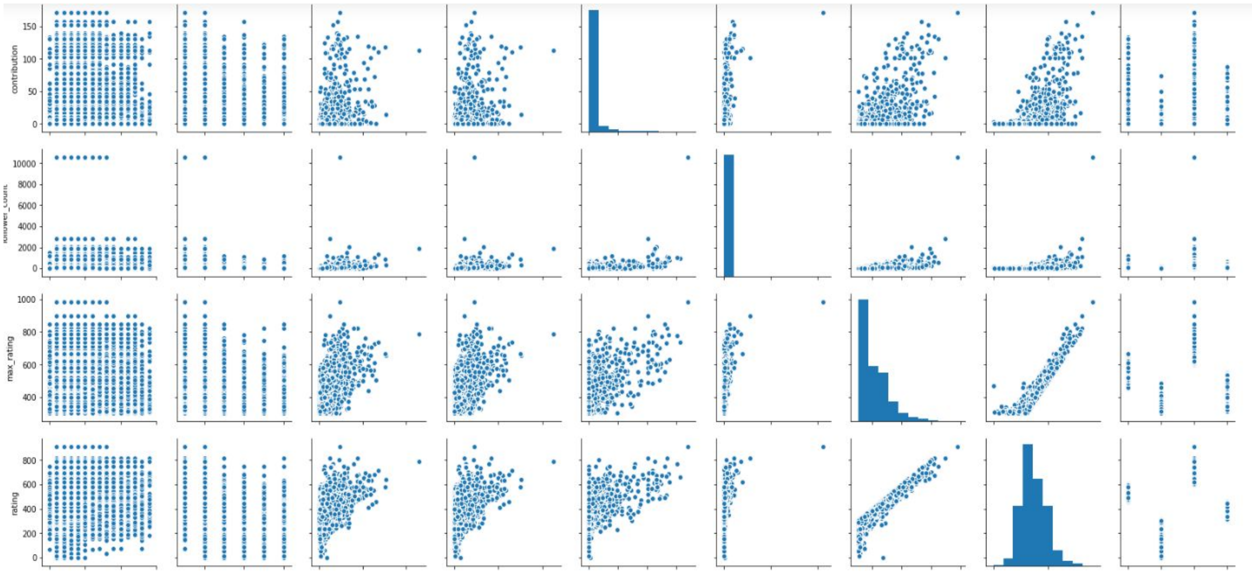
	level_type	attempts_range	submission_count	problem_solved	contribution	follower_count	max_rating	rating	rank
level_type	1.000000	-0.118299	0.247137	0.246154	0.154910	0.105787	0.323155	0.331006	-0.040756
attempts_range	-0.118299	1.000000	-0.255634	-0.254370	-0.180278	-0.111819	-0.474133	-0.537495	-0.258369
submission_count	0.247137	-0.255634	1.000000	0.997876	0.450986	0.320270	0.596288	0.546349	-0.098268
problem_solved	0.246154	-0.254370	0.997876	1.000000	0.448609	0.321789	0.596167	0.548583	-0.099975
contribution	0.154910	-0.180278	0.450986	0.448609	1.000000	0.484814	0.548499	0.485399	-0.078745
follower_count	0.105787	-0.111819	0.320270	0.321789	0.484814	1.000000	0.392347	0.342860	-0.037046
max_rating	0.323155	-0.474133	0.596288	0.596167	0.548499	0.392347	1.000000	0.941270	-0.160896
rating	0.331006	-0.537495	0.546349	0.548583	0.485399	0.342860	0.941270	1.000000	-0.075068
rank	-0.040756	-0.258369	-0.098268	-0.099975	-0.078745	-0.037046	-0.160896	-0.075068	1.000000


level_type 1.0 -0.1 0.2 0.2 0.2 0.1 0.3 0.3 -0.0
attempts_range -0.1 1.0 -0.3 -0.3 -0.2 -0.1 -0.5 -0.5 -0.3
submission_count 0.2 -0.3 1.0 1.0 0.5 0.3 0.6 0.5 -0.1
problem_solved 0.2 -0.3 1.0 1.0 0.4 0.3 0.6 0.5 -0.1
contribution 0.2 -0.2 0.5 0.4 1.0 0.5 0.5 0.5 -0.1
follower_count 0.1 -0.1 0.3 0.3 0.5 1.0 0.4 0.3 -0.0
max_rating 0.3 -0.5 0.6 0.6 0.5 0.4 1.0 0.9 -0.2
rating 0.3 -0.5 0.5 0.5 0.5 0.3 0.9 1.0 -0.1
rank -0.0 -0.3 -0.1 -0.1 -0.1 -0.0 -0.2 -0.1 1.0
level_type
ttempts_range
omission_count
roblem_solved
contribution
follower_count
max_rating
rating
rank
0.9
0.6
0.3
0.0
−0.3

Data Processing

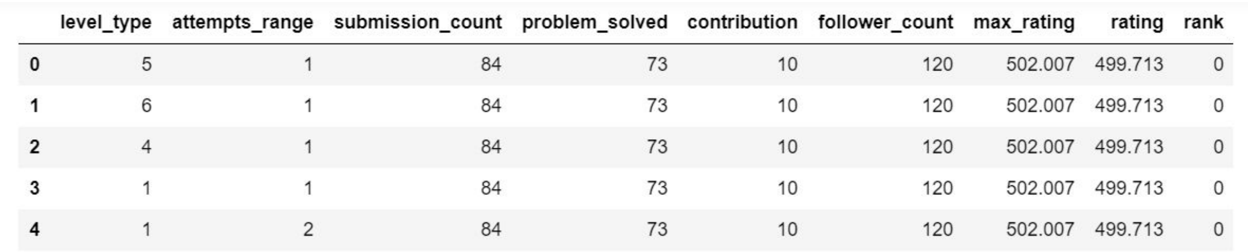
Once the above cleaning and data is processed we plot the dataset to see any evident relations for each feature.





Feature Engineering

Once all the data is cleaned and processed and performing exploratory analysis to understand the relevance of data we perform feature engineering. In this process we use label encoder *to label the level type that indicates the level of complexity of the questions and then label the rank which recognizes the performance overall of the user. We have level type data in the form of A-H and rank as Beginner, Intermediate, Advanced, Expert.*



	level_type	attempts_range	submission_count	problem_solved	contribution	follower_count	max_rating	rating	rank
0	5	1	84	73	10	120	502.007	499.713	0
1	6	1	84	73	10	120	502.007	499.713	0
2	4	1	84	73	10	120	502.007	499.713	0
3	1	1	84	73	10	120	502.007	499.713	0
4	1	2	84	73	10	120	502.007	499.713	0

Modelling

Modelling is the process of using data for making predictions which are likely to influence the future results. We went through multiple models and to name a few which brought in some kind of sense to the modelling process viz

We have used Gradient Boosting a technique for classifying problems, Random Forest Classifier and Support Vector Machine (SVM) a discriminative classifier
We also compared the result with AutoMl techniques like H2O.ai

Model Evaluation and Tuning

We used K-fold validation which is a statistical method used to estimate the skill of machine learning models. Cross validation is used to protect overfitting in predictive model, particularly in a case where the amount of data may be validated. In K cross validation data is divided into K subsets. Now the holdout method is repeated 5 times, such that each time, one of K subsets is used as test set or validation set.

Mape for Training and Testing data

Model	Test	Train
Support Vector Machine	51	54
Gradient Boost	33	45
K Fold Validation	52	

Auto ML

Automated machine learning is the process of automating the end-to-end process of applying machine learning to real-world problems.

Using H2O

AutoML is a function in H2O that automates the process of building a large number of models, with the goal of finding the "best" model without any prior knowledge.

AutoMl Model	Test	Train
H2O	48%	51%

Model Selection

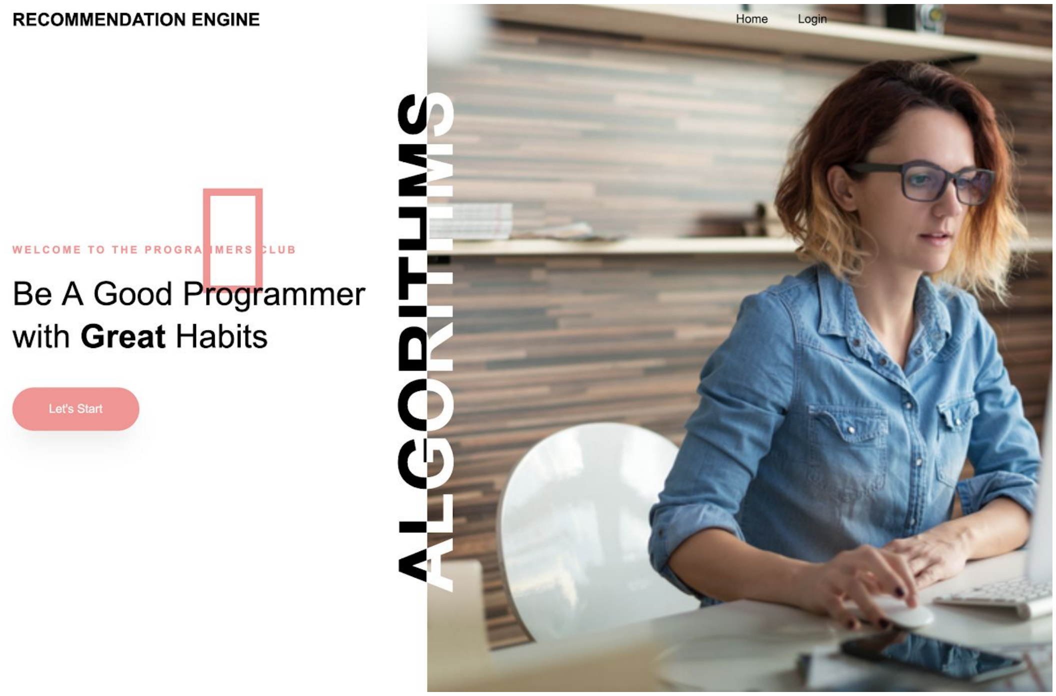
On the basis of above evaluation and tuning evaluation we see that Support Vector Machine provides the best result for our model.

Model Deployment

We pickle our model by creating a pickle file. Pickle is the standard way of serializing objects in Python. One can use the pickle operation to serialize your machine learning algorithms and save the serialized format to a file. Later you can load this file to deserialize your model and use it to make new predictions. Thus we created a pickle file which is then integrated with a flask application which is then deployed on heroku to host and run the application.

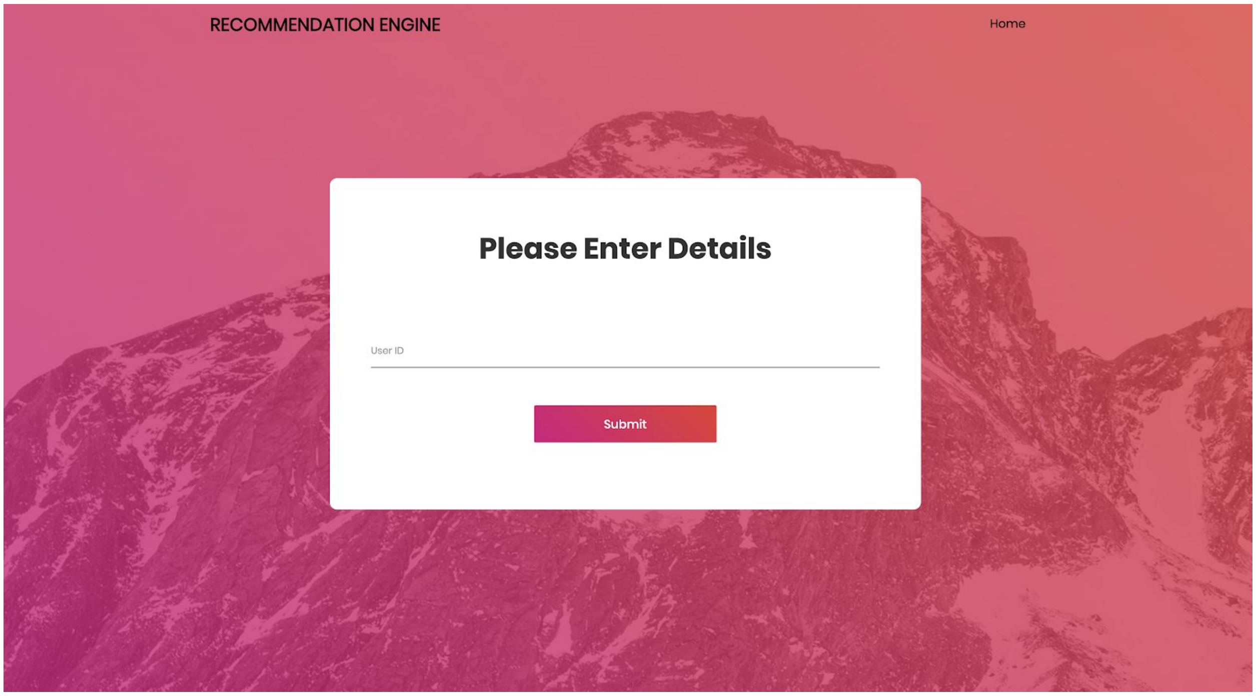
1. **Language**: Python, Html, css, javascript
2. **Web Framework:** Flask
3. **Container**: Docker
4. **Web Platform**: Heroku

USER INTERFACE -

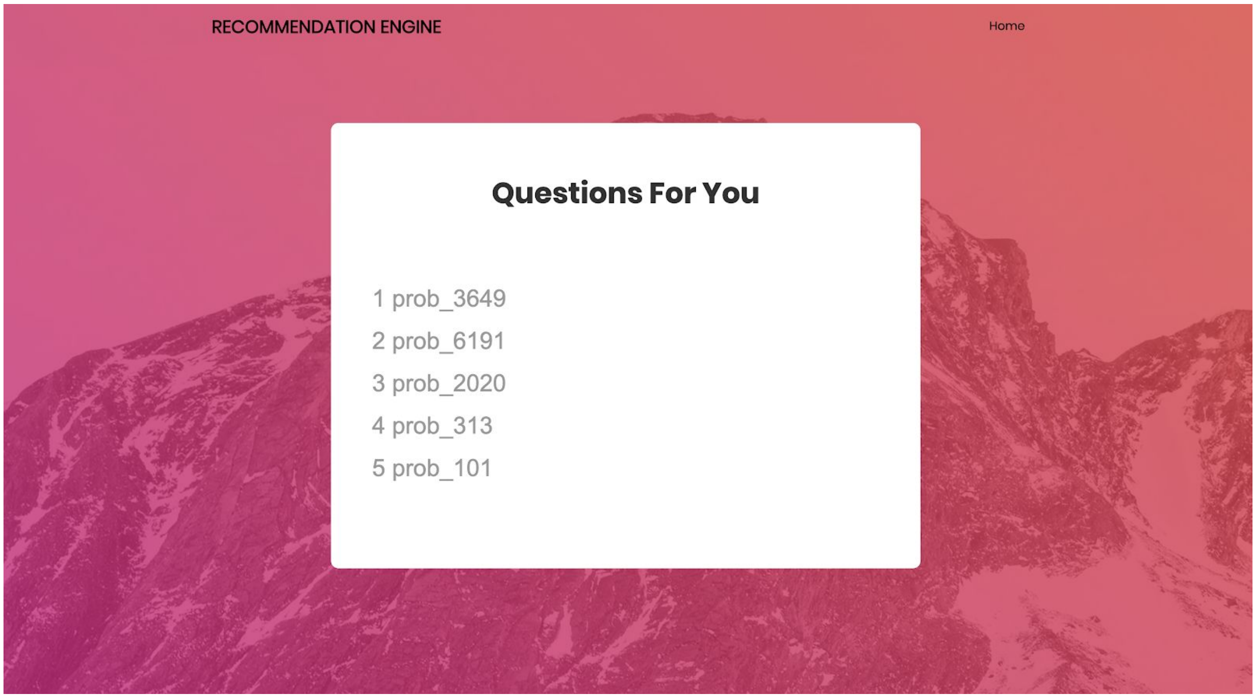


Be A Good Programmer
with **Great** Habits

Our application expects you to log into the application and look input user id. The input is then looked up in the excel file and given as input to our model. The model then based on multiple features provides list of questions as output that it infers as the user can solve.



Please Enter Details



Questions For You

```
1 prob_3649
2 prob_6191
3 prob_2020
4 prob_313
5 prob_101
```

Reference

https://dzone.com/articles/using-an-automl-h2o-model-to-predict-attrition-and

https://medium.com/analytics-vidhya/gentle-introduction-to-automl-from-h2o-ai-a42b393b4ba2

https://stackabuse.com/scikit-learn-save-and-restore-models/

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

https://www.heroku.com/

http://flask.pocoo.org/