REPORT

Files:

The zip file contains two ipython notebooks namely '***scripts.ipynb***' and '***predict.ipynb***' files. There is also a ***'predictions.csv'*** and a ***'predictions_labels.csv'*** files which contains the prediction for queries in the '***test.csv'. 'predictions.csv'*** has (user_id,problem_id,prediction) entries and ***'predictions_labels.csv'*** has just predictions which correspond to each entry in test.csv.

'***scripts.ipynb***' file consists of scripts to analyze the datasets and various snippets for plotting graphs and cleaning data.

'***predict.ipynb***' file contains a machine learning model which can predict whether a user will solve a problem or not.

NOTE: The ipython notebooks are also stripped to python scripts namely “**scripts.py**” and “**predict.py**”

Steps to reproduce results:

The ipython notebooks are well commented and intention of each step is described. Several computationally intensive objects are written to files and later read when needed. The results can be reproduced by running these ipython notebooks or by running the corresponding python scripts of these notebooks.

Technology, tools and language used:

Language	:	Python
Tools	:	Jupyter Ipython notebook, Sublime editor, LiberOffice calc
ML libraries	:	Scikit-learn
Frameworks	:	Pandas, Numpy, Seaborn

Answers to Questions:

Steps of these question can be found in '*scripts.ipynb'*****

Q1) **Analyze average number of attempts it took for users to successfully solve problems at Hackerearth?**
Ans: 14.1841835918759

Q2) **Did they struggled irrespective of level of difficulty of problems or they only struggled in hard level problems??**
Ans: The level of difficulty is available in the problems.csv file. There are five level of difficulty namely E, M, M-H, E-M and

H. The avergae accuracy for each level was computed. Accuracy is direct measure of number of successful submissions against total number of submissions. This clearly is depicts how much users would have struggled for each of these level. The following figure explains the average accuracy of each levels. Clearly the accuracy for Hard problems is low and Easy problems are high.

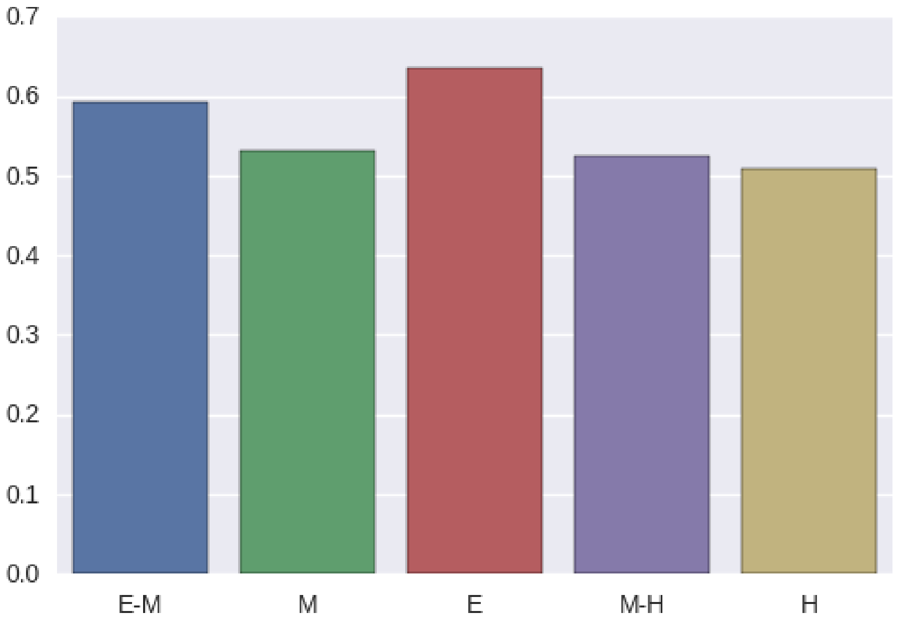

Fig 1: Average accuracy versus level of difficulty

Q3)**Calculate the average percentage of time improvement on successful submission over unsuccessful/partially accepted attempts.**

Ans: The submissions.csv file has (user_id,problem_id) to identify submission for a user-problem pair. It has been noted that few user got successful submissions in first attempt. Also there are user who couldnt get the solution accepted. Such users where ignored from the computations. The users who got partially accepted in the begining and got accepted after successive submission where considered. Also it was found that the accepted solution has got a higher running-time than the partially accepted solutions. These combinations where ignored. The rows considered for computation where the ones in which the accepted solution is of lower execution time while the partially accepted solutions were higher.

A few the computed values are shown below:

User_id	**Average % improvement**
967552	405.665271192
1178898	8.70209759626
1130935	343.089779301
1034752	17.3110859312

NOTE: Some of the user have got a huge % improvement. It was observed that several partially accepted execution times where very large compared to the accepeted. E.g: For user 1130935 there are cases where the python submission had a runtime of 105 ms for PAC and 5.18 ms for AC. These are huge % increases.

Prediction Model:

The given dataset was thoroughly analysed and found that training and test data are given. The training set consists of users,problems and submission history. The test set consisted of users and problems and a file test.csv which contains user-problem pairs. A model was created which could predict whether a user would solve a given problem or not.

The submissions file in the training set was cleaned and modified to (user_id,problem_id,solved). Solved column contains '+1' if the user got the problem accepted and '-1' if the user got it partially accepted.

The model was then trained using the feature of user and problems against the target label 'solved'.

The user features and problem features had to be cleaned and empty cell were filled using appropriate values. There are several categorical features which were encoded. A Random forest classifier was used because it appeared to perform well with categorical features.

NOTE: An accuracy of 97.49% was achieved on the training set.

Findings on Trends, Visualisation and/or Data points:

1)Which tag1 category of questions are most solved???

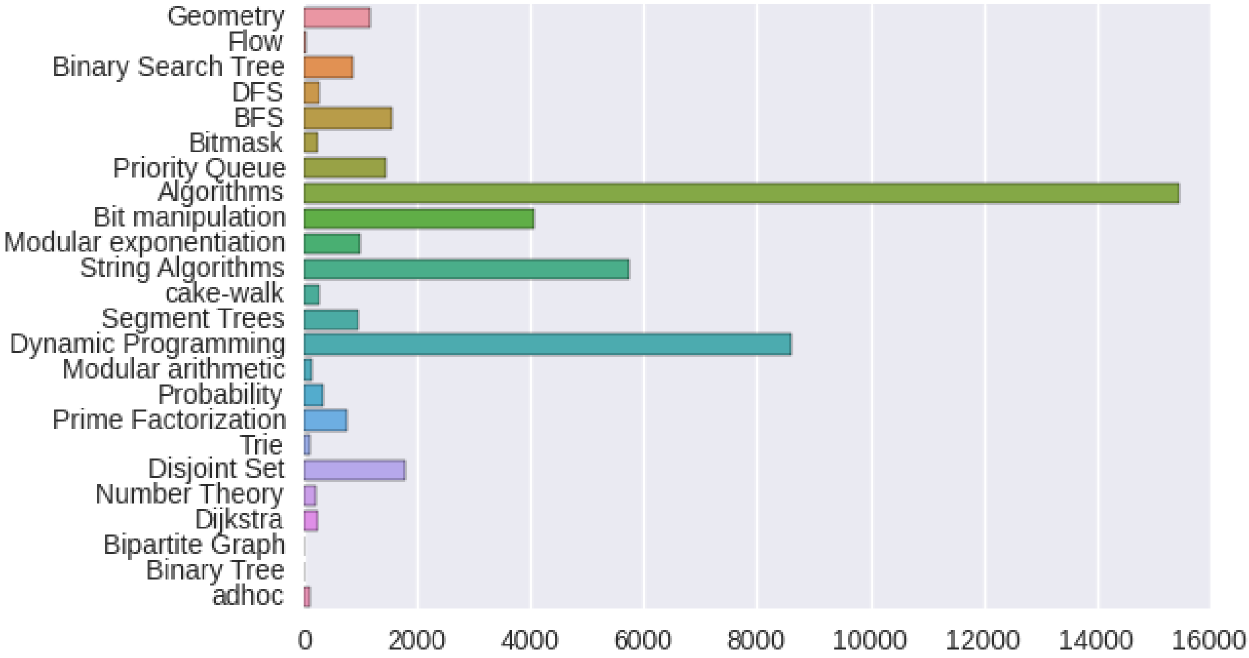

Fig 2: Category(tag1) of question versus solved count

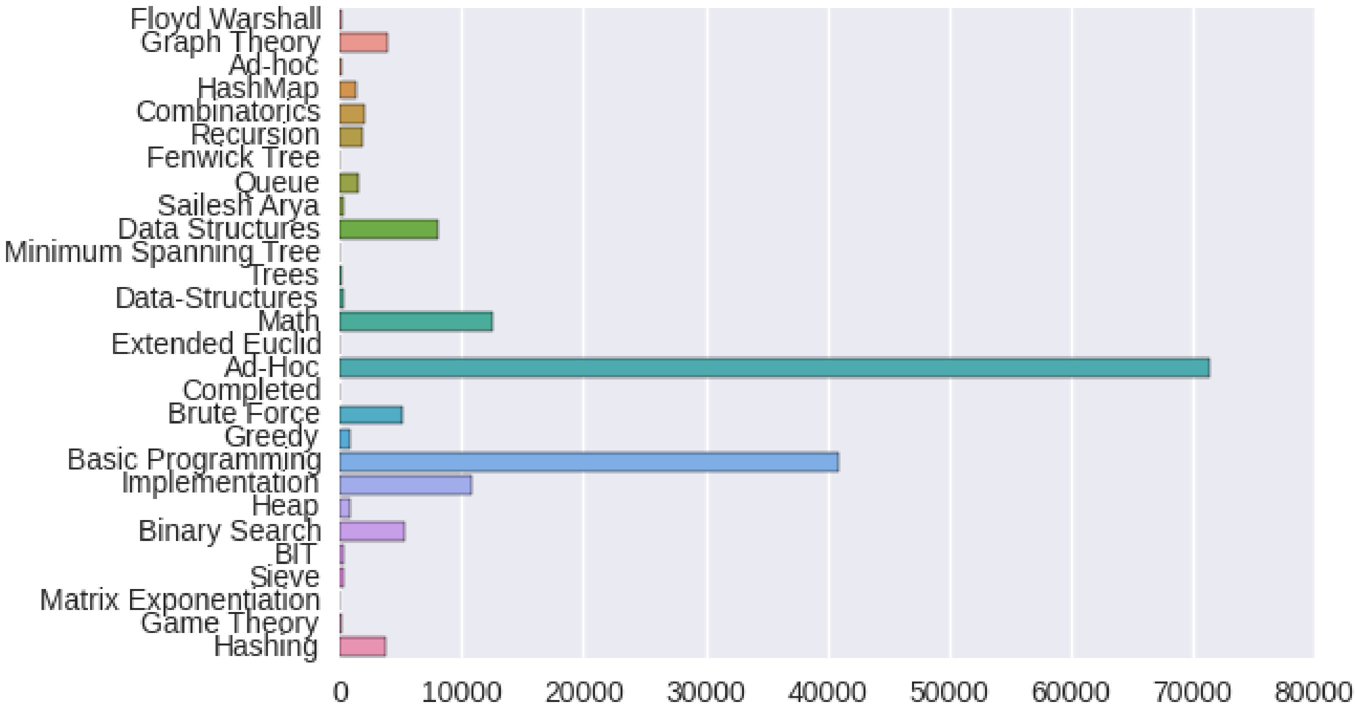

Fig 3: Category(tag1) of question versus solved count

Users seems to solve more of Algorithms, Ad-Hoc ,Dynamic Programming and Basic Programming.

2)**Which tag2 category of question have highest error rate????**

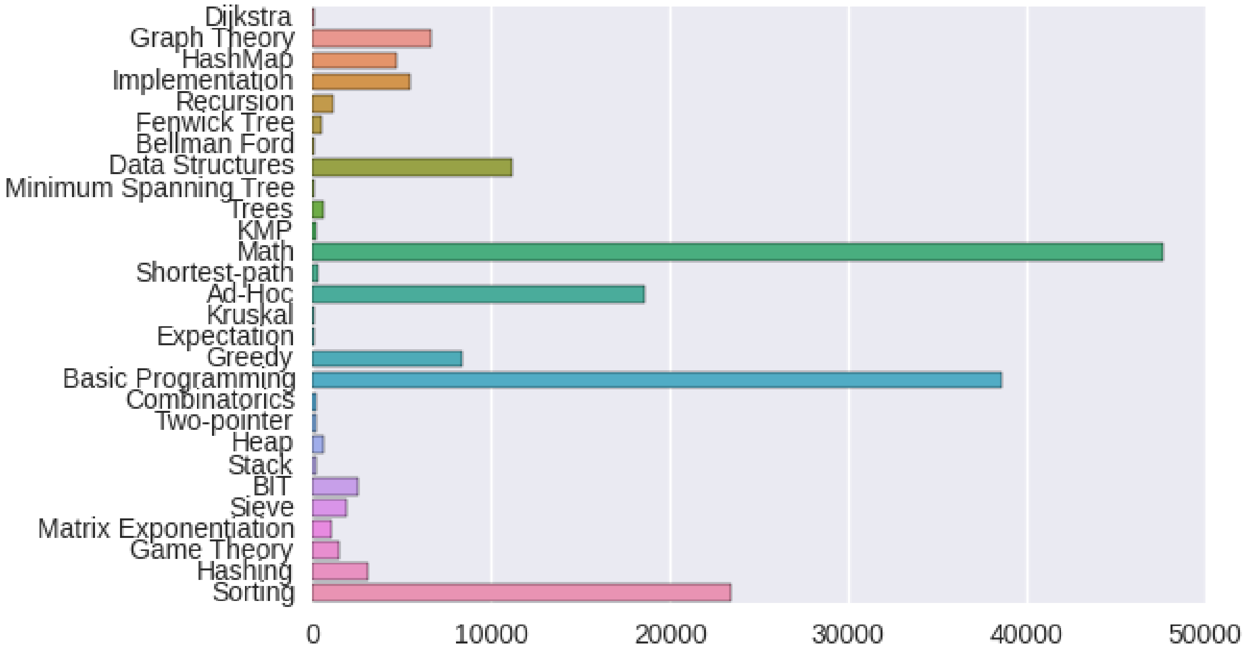

Fig 4: Category(tag2) of question versus error count

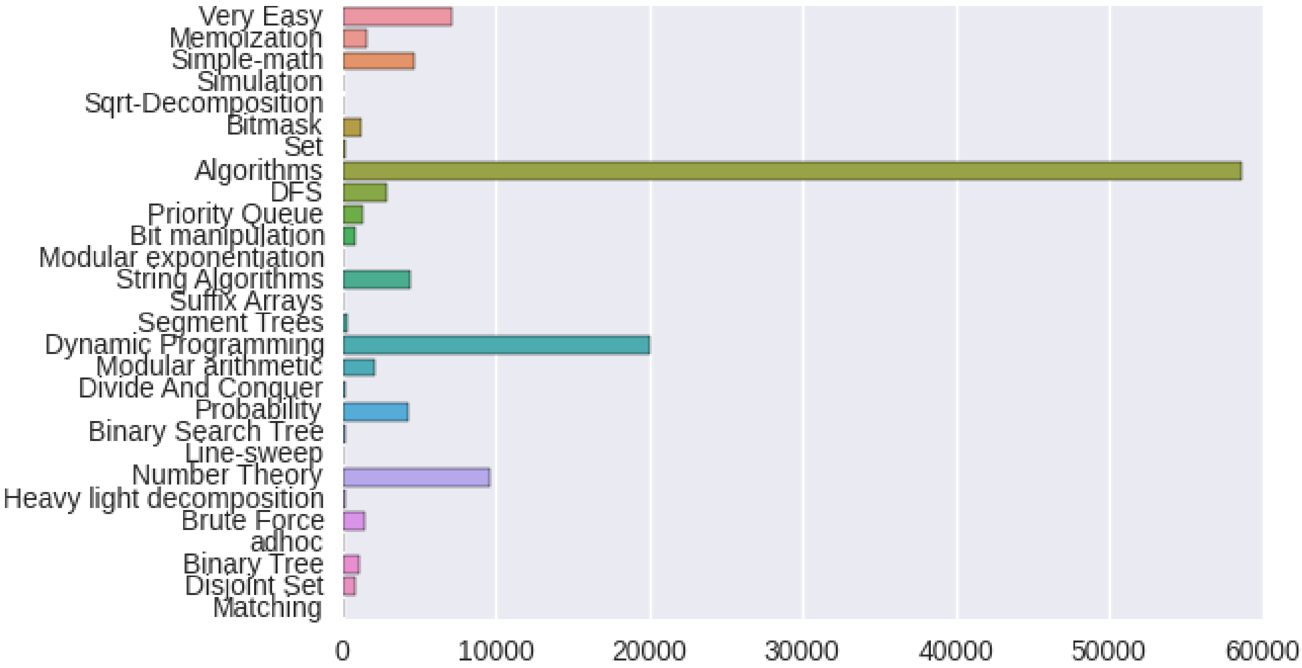

Fig 5: Category(tag2) of question versus error count

The Algorithm category seem to have huge erroreous submissions.

3)Which programming languages are more used by users???

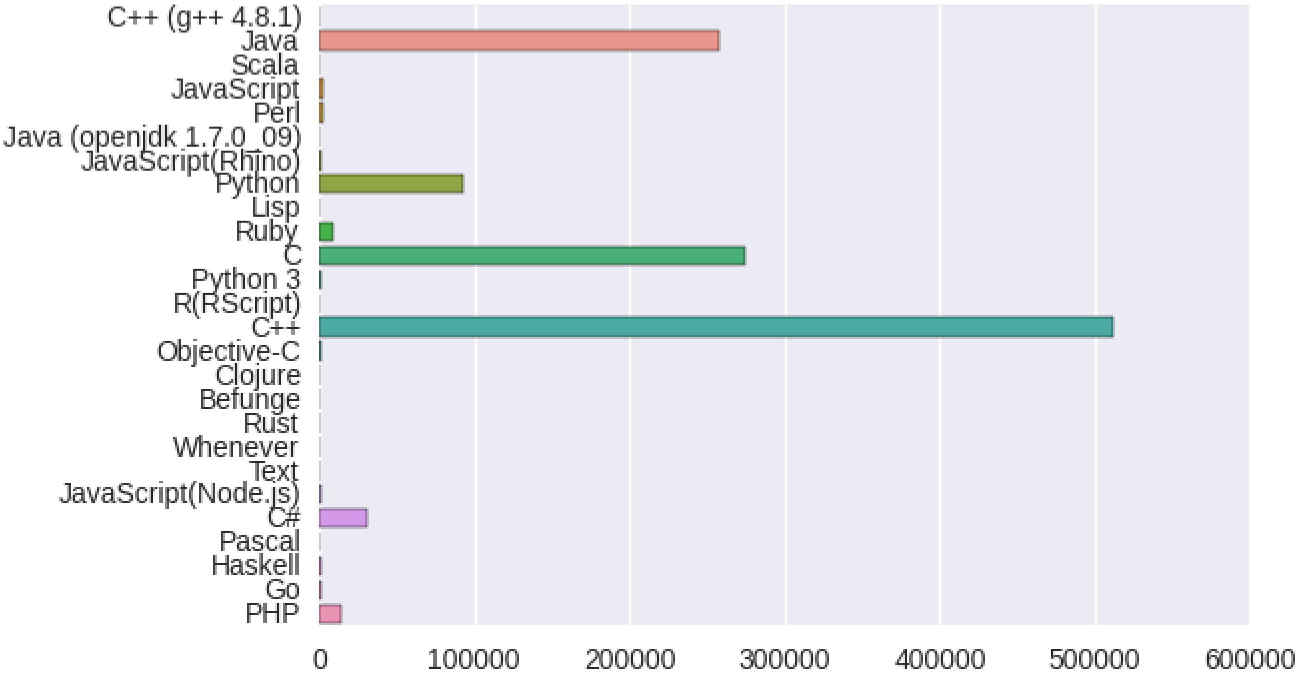

Fig 6: Programming languages versus number of submissions

C++ seems to be the most widely used language.

4)What user type uses hackerearth more???

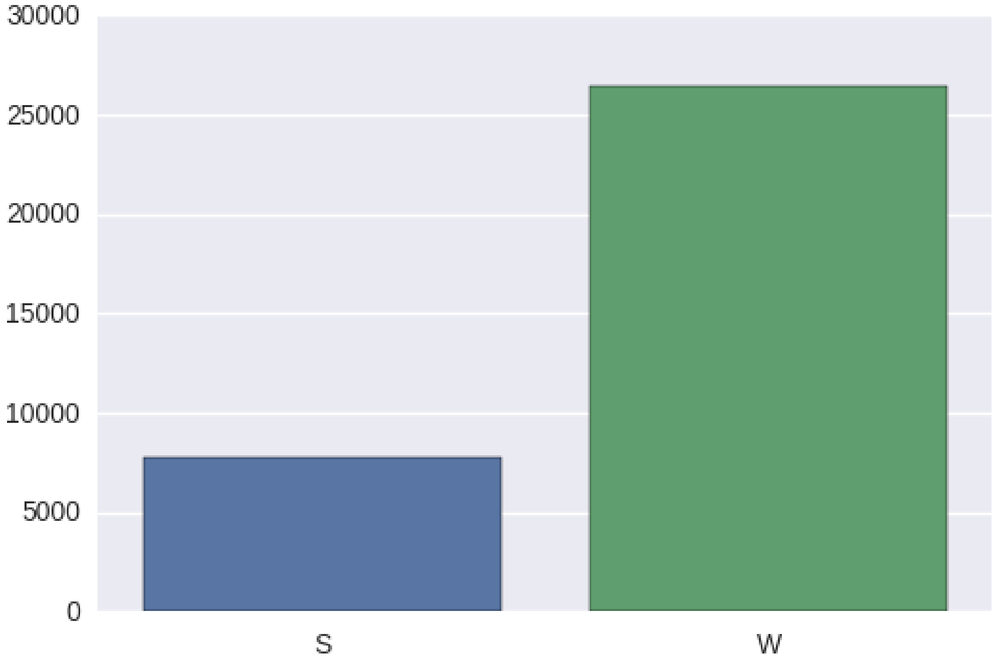

Fig 6 : user type versus count

It appear the user type W (i think it means 'Working') uses hackerearth site more.