



Competitive Programming Vestige Using Machine Learning

Ajay Dharmarajula^(✉), Challa Sahithi, and G. S. Prasada Reddy^(ID)

Department of Computer Science and Engineering, Vardhaman College of Engineering,
Hyderabad, Telangana, India
ajaydr16@gmail.com

Abstract. Competitive programming improves our problem-solving ability. It helps in writing the source code of computer programs that help in solving given problems, and the majority of the problems are mathematical or logical in nature. In view of its staggering and different nature, programming requires a particular level of expertise in the examination of estimations, data structures, science, formal reasoning, and related tasks like testing and investigating. In light of the growing regard for expectations for programming, there exist different genuine programming platforms like HackerRank, CodeChef, CodeForces, Spoj, etc. where students can practice and work on their competitive programming skills. Monitoring the progress on these different platforms becomes hectic as they have to manually check each one. Also, there is no tool that helps in predicting their future scores based on their current practice. Another issue is that if the organisations or institutions wanted to monitor their student's progress, it would be tougher as it would have to be done for each student manually. This work will help the students, as well as the organisations or institutions, maintain a proper portal with data to monitor their progress, by which students can improve their competitive programming skills, saving a lot of time compared to the time taken to do this monitoring manually.

Keywords: Machine Learning · Web Scraping · Competitive Programming

1 Introduction

There is a huge population of youth who are the future of our country but India lacks the skill factor in youth. No doubt everyone is getting educated but they always lack in skills required by industry. In the software industry, the main skill required by each and everyone is Competitive Programming. Organizations/Universities always focus on improvising students' skill sets but in competitive programming, regular observation is required. But this observation when done manually takes a lot of time, and effort and becomes a hectic task.

Foreseeing the scholastic exhibition of students has generally been for quite a while, a sharp area of interest for some specialists. Various investigations have been directed with the goal to measure the connection between the scholastic capability of a student

and the variables influencing it. Inspecting the components liable for influencing a student's insightful execution wouldn't simply assist us in exploring a better appearance frameworks yet also having the option to help in decreasing the overall drop-out rates among energetic students. Progressions through electronic learning give us overflowing measures of information to see what students realize what elements mean for their scholarly presentation. Online learning archives of the information contains data that could assist us with opening recent fads in educating furthermore, draw bits of knowledge about the current nature of schooling and how could be given a better degree of training than students.

"Competitive Programming Vestige using Machine Learning" is a data analytical paper that uses machine learning methods and will automate these observations and also predicts future results. This paper will scrape the data from every coding platform that a the particular student has registered, store it in the database and then analyse the data as a whole to the student as well as to the organization. This analysis includes future progress, leader board, and many such useful projections for students/organizations to be taken in order to excel. These days, there exists a downpour of web-based programming conditions, for example, programming sites, internet programming instructional exercises and cloud-based Integrated Development Environments (IDEs) giving their clients different apparatuses to sharpen their programming abilities. These programming conditions contain a lot of information about the students' critical thinking history along with data about the nature of programming issues. By examining the previously solved problems of the students, we can endeavour to anticipate the projection of the student. To lead a prescient examination, we gathered information from different competitive programming spaces. Datasets were separated from CodeChef, codeforces, interview-bit, spoj. These are competitive programming sites that give programming difficulties of fluctuating degrees of trouble to its clients and these are additionally involved by many organizations for recruiting possible representatives. These datasets incorporate a similar list of capabilities for assessing student execution. These datasets were independently dissected to derive to foresee the student's progress in individual stages.

2 Related Work

There has been a huge load of studies done to predict understudy's academic execution using information mining procedures. The explanation of information mining has turned into a generally famous instrument for surveying a student's execution is on the grounds that, it gives unique instruments to break down the information, and thusly helps us in finding recent fads and connections.

Crafted by R. Kaushal and A. Singh [1], clarifies the significance of programmed appraisal of programming tasks. The creators have fostered a programming entry that means assessing the submitted code naturally and subsequently, grade the understudy in view of their presentation. Their work definitively demonstrates that the execution of such apparatuses in a learning climate, can work on the understudy's execution as far as effectiveness, consistency and precision.

In a comparable work, Medha Sagar, Arushi Gupta and Rishabh Kaushal [2], utilized datasets from HackerEarth and its entryway to examine and survey understudy's

execution. The introduced paper draws its motivation from the recently talked about examinations. We have additionally done a relative examination of characterization strategies to choose the most fitting and effective calculation for foreseeing the understudy execution. However, aside from anticipating the presentation, we have fostered an undeniable full-stack application which can be promptly use by students and institutions to get a clearer outcome.

The work of Timmy Devsia, T.P.Vinushree and Hegde Vinayak [3] have accepted the understudy's scholastic history as information and given understudies' impending exhibitions based on semester. Their work definitively proves the accuracy of using the Naive Bayesian algorithm.

In a similar work, M. Sivasakthi [4] worked under data mining which deals with a knowledge flow model for every one of the five classifiers and it grandstands the significance of Prediction and Classification based information mining calculations in the field of programming schooling and furthermore presents a few promising future lines. It could carry advantages and effects on understudies, instructors and scholastic organizations.

3 Proposed Approach

3.1 Data Set

To lead a prescient examination, we gathered information from different competitive programming spaces. Datasets were separated from CodeChef, codeforces, interviewbit, spoj. These are competitive programming sites that give programming difficulties of fluctuating degrees of trouble for its clients and these are additionally involved by many organizations for recruiting possible representatives. These datasets incorporate a similar list of capabilities for assessing the understudy execution. These datasets were independently dissected to derive to foresee the understudy's progress on individual stages.

3.2 Preprocessing

Information Pre-processing incorporates eliminating boisterous information or exceptions. Information is gathered from various coding stages like Codechef, Codeforces and so forth through Web Scraping. One of the means in Data Pre-processing is bringing in libraries. Consequently, for the dataset as web scraping is required, libraries like Beautiful Soup in python helps by supporting Web Scraping. Libraries like NumPy and Pandas are likewise expected for logical computing, manipulation and investigation. For factual modelling(regression) the Scikitlearn library is utilized in Python.

3.3 Supervised Learning

Supervised Learning in Machine Learning is a technique through which a machine gains from the it is as of now marked to prepare information which. It is utilized to derive a capacity by learning through the given info information that assists with deciding result

for the new inconspicuous information. It assists the prescient models with spreading the word about expectations from the information. Regression and Classification are the two strategies that go under Supervised Learning. Regression method is utilized to anticipate nonstop factors like rating. It is essential to utilize supervised learning since it helps the machine by giving a learning experience that aids in ongoing forecasts. These encounters are likewise useful for the improvement of the presentation. Linear Regression: Linear regression is the most widely recognized calculation that is utilized in Predictive Analysis to foresee consistent factors. Linear Regression is a supervised learning calculation. Displaying the connection between one ward variable and at least one autonomous factors by fitting a linear connection between them is utilized. There are two sorts of linear regression calculations in view of the quantity of free factors utilized. They are Simple linear regression and Multiple linear Regression.

The linear condition would be $Y = a_0 + a_1 * x$ where y is a dependent (response) variable whose worth is to be anticipated and x is a free factor which is utilized for prediction. Here a_0 is the capture and a_1 is the incline of the regression line. Here rating is the reliant variable which should be anticipated.

4 Proposed Model

4.1 Experimental Setup

Web Technologies and Python Frameworks assumes significant part where utilizing of moving tech for our application builds usefulness and productivity. A Full Stack design is utilized to put together everything in a coordinated manner. Using front end as Angular builds Browser streamlining where Angular backings respond, switch Dom where delivering is done at client side, stacking of pages takes less time as it permits parts delivering concurrently. This way part of time will be saved rather than page stacking and page reviving. Backend we involved Flask framework for building an API based cooperation among frontend and backend. Use of flask gives a light weight to the application and APIs can be deployed without any problem. At last, we involved Google's Firestore datafactorr putting away all client subtleties in a key-value style likewise their present scores or evaluations according to their enrolled platforms. This way entire full stack application will be working autonomous to one another and working in a coordinated manner (Fig. 1).

Angular is an improvement stage, based on TypeScript. We construct the total front utilizing Angular where we isolated the application into different parts, designs, pages, factors, modules, and resources. Fostered a login page as the point of arrival once the utilization land on. Users can give his/her credentials to log in to it or probably can tap on register to enrol another user.

When the user sign in the application sidetracks to the user dashboard which we have delivered the page with a huge number. Here every one of the information of different platforms like codechef, codeforces, spoj and a lot more are being pulled from the firestore database however in-fabricated API calls and the response is being delivered into individual parts then, at that point, populated to the dashboard. Same way the predicted graphs additionally being gotten as a response from backend and populated

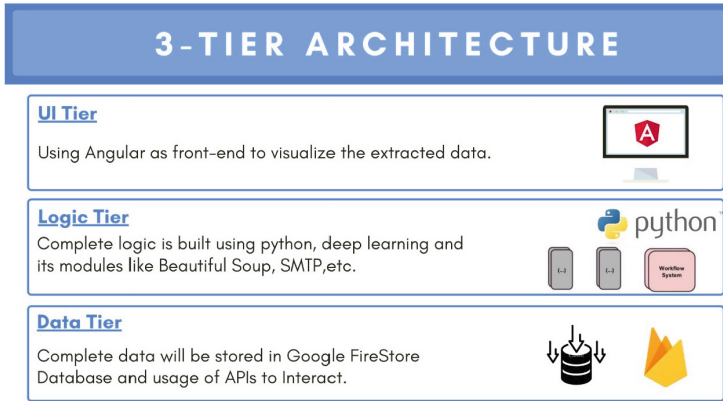


Fig. 1. Three Tier Architecture

here on dashboard. Aside from those we have created different extra highlights like top pioneer board information in a type of appropriate organized table.

Same way we have fostered an administrator dashboard where administrator can deal with every one of the students of their organization fetched directly from firestone database at a solitary page and separately can get to any profile. Above all can perform numerous sorts of information scientific operations like greatest, least, looking, arranging, looking through in view of a individual platform, looking through in particular year, picking just a specific year, block unblock a user and parcel of such insightful cycles on a solitary page window (Figs. 2, 3 and 4).

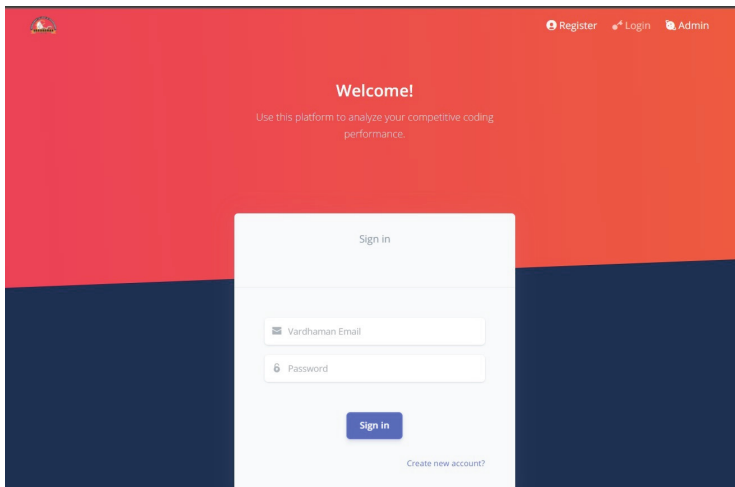


Fig. 2. User Login

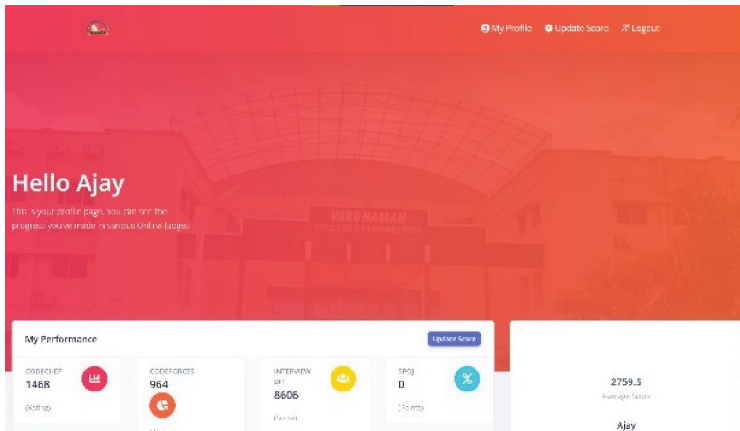


Fig. 3. User Login

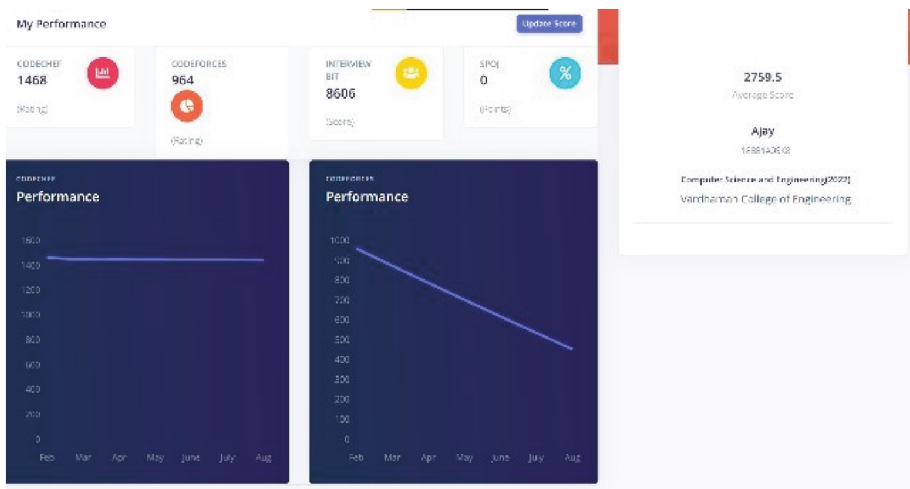


Fig. 4. User Dashboard

A web framework is an engineering containing instruments, libraries, and functionalities appropriate to fabricate and keep up with huge web tools utilizing a quick and productive methodology. They are intended to smooth out programs and advance code reuse. To make the server-side of the web application, you want to utilize a server-side language. Python is home to various such structures, renowned among which are Django and Flask. Python Flask Framework is a lightweight miniature structure in view of Werkzeug, Jinja2. It is known as a miniature system since it expects to keep its center usefulness little yet normally extensible to cover a variety of little and enormous applications. Cup Framework relies upon two outer libraries: The Jinja2 format, Werkzeug WSGI tool compartment. Utilizing flask framework, we have fabricated APIs for individual platforms through which utilizing the web scraping idea we are scrapping all

the user information or action from the hour of enlistment to work now. Additionally, we ensure that we diminish the quantity of APIs since more the end points are there more will be the traffic and weighty the application becomes. So, we considered just to make just a solitary API for scrapping information and one more API for anticipating the information.

The information being anticipated is finished by considering the dataset as under-studies recently addressed or endeavoured programs and their appraisals. In light of considering and taking that dataset, we have utilized machine learning calculations to anticipate future evaluations or score assuming the understudy or the user go on similarly as now. Given the prediction, user will get to know the amount he/she wants to improve and create. Every one of the information that is caught is shipped off the firestone data set for putting away this rejected information in a key-esteem design. Later this information is sent as a response to the UI for example frontend Angular (Fig. 5).

ROLL NUMBER	NAME	BRANCH	YEAR	CODECHEF	CODEFORCES	SPOJ	INTERVIEWBIT	STATUS
18881A05P3	Jahnavi Suram	CSE	2023	1459	0	0	889	Unblock
18881A05G2	Chidvika	CSE	2022	1549	1392	0	0	Unblock
18881A05K6	Challa Sahithi	CSE	2022	1473	1065	0	2730	Unblock
18881A05A0	Jahnavi Nambi	CSE	2023	1403	955	0	4383	Unblock
18881A05K7	Pavan Kumar	CSE	2022	1508	393	0	1428	Unblock
18881A05G7	Gurha Teja	CSE	2022	1488	1371	0	1315	Unblock
18881A05K8	Ajay Dharmarajula	CSE	2022	1468	964	0	8606	Unblock
18881A02A3	Sanjana	EEE	2022	1336	360	0	2242	Unblock
18881A05O2	Vivek Assam	CSE	2022	1449	1392	0	9376	Unblock

Fig. 5. Admin Dashboard

We have used Heroku for deploying the backend APIs. Utilizing Heroku CLI it turned out to be not difficult to deploy by coordinating GitHub is the repository and every one of the frequent changes is effectively being pushed to the Heroku deployment. With that end, point is made for APIs through which generally GET and POST-based API callings are done without any problem.

We utilized firestore to store every one of the information of the application. It gives many defaults in-constructed APIs through which information can be embedded, erased and fetched. Firestore APIs were utilized to push the user enlistment information and later their platform scores and evaluations including their critical thinking action. We made two documents (for example called tables in SQL) where in one we put away every one of the user’s essential subtleties like name, year, institute name, email, password

and usernames of the platforms they enlisted in and in another document, we put away every one of the information that is being scrapped from them regarding online judges.

At long last, we deployed and hosted the entire application in firebase utilizing firebase CLI commands. Through this organization any place the institute and regardless of where the student is available all can undoubtedly access with a solitary snap of this facilitated URL and they can further develop their range of abilities by remaining fixed on this particular application.

4.2 Proposed Algorithm

The algorithm is a collaboration or set of rules to be gone on in calculations or other decisive reasoning activities, especially by a PC. We have proposed two algorithms for this application one is user algorithm and another one is admin algorithm.

User Algorithm. User Algorithm is the algorithm or the set of process that will be performed on the user-side of the application starting from user registration till user logouts. Before going to the user dashboard frontend angular will authenticate whether user is logged in or not, based on that decision it will redirect to user login panel if the user is not logged in. Again, based on users entered credentials it will redirect to dashboard (Fig. 6).

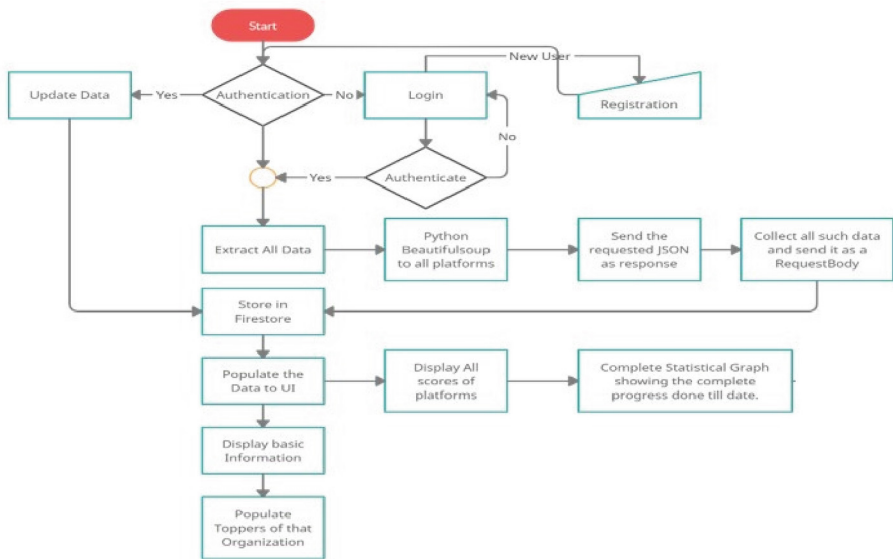


Fig. 6. User Algorithm

In the user dashboard the whatever platform details given while registering the account all those details whether rating or points as per the platform will be populated on to the UI from the firestone database. All the data are completely stored and fetched from

Google's firestone database. Also based the details fetched from the database this application will plot a graph about future six months prediction of that candidate based on his previous attempts on that platform. These complete six months prediction is predicted from the backend that we have developed though flask framework, detail information about the prediction and backend operations will be discussed in backend algorithm below. Apart from prediction application also shows at which rank that student is on as per others students and the complete top leader board of the students of same college. Based on all these projections a user can determine where he/she stands among others and what his/her progress will be in future months if the flow goes like that.

Admin Algorithm. Admin Algorithm is the algorithm or the set of the process that will be performed on the admin side of the application starting from admin login till admin logouts. Before going to the admin dashboard frontend angular will authenticate whether the admin is logged in or not same like the above user algorithm, based on that decision it will redirect to the admin login panel if the admin is not logged in. Again, based on the admin's entered credentials it will redirect to the dashboard.

In the admin dashboard whatever data that is been entered by students of that college will be populated from the firestone database through FETCH API. API will send the response as JSON and all the collected data is filtered and populated to the UI. Here all the student's detail along with their platform performances are listed in a tabular form. This tabular form of data is completely filtered as per the admin requirements. Example if the user just wants to know a specific list of data only of a singular platform, or search from only a particular year and many such filters. As an admin, admin can click on or go to any user in their organization to check their progress that has done and predictions. Admin has the right to block a user temporarily in the organization and can do many such admin rights that are available. With such algorithm admin can completely monitor their students' progress by just with a click which saves a lot of manual hours of work time.

Backend Algorithm. Not until unnecessarily at some point previously, server- side conveying, or back-end web headway, was the acknowledged strategy for making sites and web applications. You visit a page, send a sales for content, the server processes this sales and makes a response that is sent back to your program. Exactly when a site page renders server-side, all of the cycles drew in with making a HTML page that your web program can understand are dealt with on a far-off server working with the site or web application. This join addresses data sets for information and dealing with any reasoning that your the web application requires. While the faraway server is involved working, your web program is idle, believing that the server will wrap up dealing with the requesting and sending a response. At the point when the response is gotten, web programs translate it and show the output on the screen (Fig. 7).

A complete backend is created utilizing the Flask framework and this is the place where every one of the intelligent activities happen. We fabricated the REST APIs through which front and backend can interface and trade data according to necessity. We have different capacities running at the backend yet significantly we have partitioned the backend into two kinds one is web-scraping and one more is the expectation of gathered data.

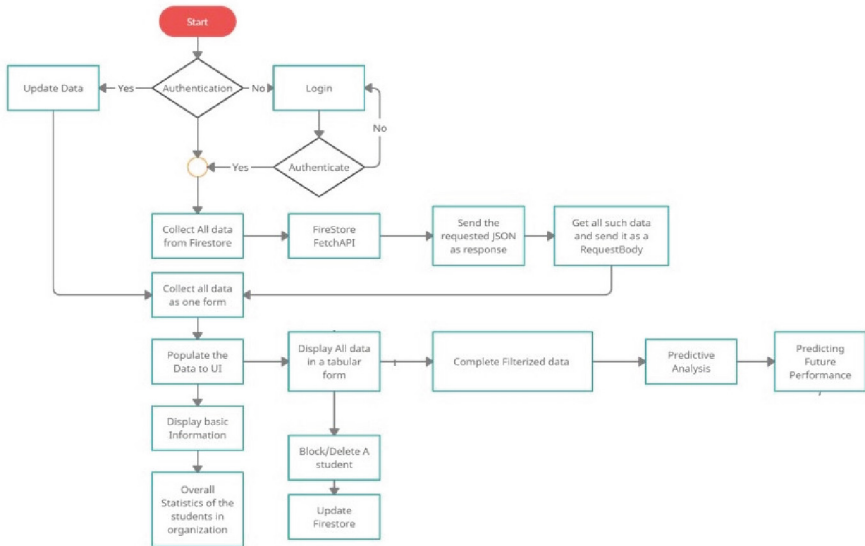


Fig. 7. Admin Algorithm

First coming to the web-scraping part we have significantly involved the BeautifulSoup module of python for accomplishing this. By introducing this module and passing the separate client dashboard URL for individual website we have accumulated the total source code of that page. Later we saw that each field, each worth, or data is available in different classes, div, length, headers so we noted down anything labels that are been implanted to and utilized those label names to get individual data of that comparing esteem. Then, at that point, we have put away every one of the gathered qualities put away and changed over the entire data gathered into a JSON organization and sent back to the frontend and furthermore this gathered data is likewise been pushed to firestore database. As it is an API based cooperation so when the solicitation got from frontend about specific stage and given username. Backend will do the mechanization and sent back the gathered data as a reaction to the frontend.

Presently the subsequent significant part is expectation, precise forecast about rating or score is profoundly unthinkable on the grounds that those last qualities relies upon different elements like what sort of challenge it is, the number of understudies are taking part in it, the number of individuals tackled that issue, the number of settled to some degree, the number of neglected to saw and a great deal of such factors decide the score of any challenge. So, these variables can't be anticipated as it is just can be taken once the challenge is finished. However, we have utilized specific boundaries and attempted to foresee the following a half year expectation of the client. This way client will get to know where he/she is the present moment and will be roused to work more to get a slanted diagram on their advancement. Presently coming to the expectation calculation, we have utilized direct relapse since there is just a single free fan actor through which forecast should be possible. Utilized straight relapse and made a dataset comprising of clients past addressed measurements and past appraisals this way utilizing the connection

between the gathered qualities next rating is anticipated, in view of anticipated rating next rating is anticipated this way we anticipated next a half year rating. This way client can comprehend on the off chance that they go on similarly as they are doing now, what will be their advancement by the following a half year. Indeed, even executive can screen and figure out who are those clients and can take an individual consideration for them with the goal that their the presentation will be gotten to the next level. At last, we have facilitated all the backend code on the Heroku server. Utilized the gunicorn module in python to change over all the HTTP demands into real python code and utilized the end-point of it to settle on REST calls from the frontend.

4.3 Proposed Model Objectives

- The main objective of our paper is to collect the data of students from various coding platforms they use and then analyse it using Machine learning methods to suggest them the improvements which need to be taken to master the skills without losing interest.
- Students often procrastinate their work by which they lose interest in learning the skills after some time. Hence, they need continuous guidance that is only possible through organization and our paper helps the organization by giving suggestions.
- Our paper predicts the future target of the student based on their daily performance which helps the practice to go smoothly.

4.4 Proposed Model Outcomes

- Complete Data Analysis of the data scrapped.
- Improved Results in the student's performance.
- Time reduction for organizations' efforts.

5 Results and Discussion

Assuming that you're a Computer Science understudy or a coding devotee, odds are more you've heard people examining their Competitive Programming abilities and rankings and accomplishments in different coding difficulties or challenges. Furthermore, really, The competitive Programming ability is one of the couples of abilities that put you aside from the group, and in this manner furnish you with an edge over others during positions or other vocations amazing open doors. Numerous eminent tech monsters including Meta, Google, Amazon, and so forth altogether think about the competitive programming abilities of the competitors and even enlist the up-and-comers through different competitive programming challenges.

Competitive Programming is an activity to step up the Programming and Data Structures and Algorithms abilities through tackling various certifiable programming issues under specific pivotal limitations including time limit, memory requirements, time and space intricacy, and so forth Understudies need to concoct an advanced arrangement inside as far as possible for the given programming issue in your favoured language and the code should finish all the necessary assessment cases. The best part is here you

rival different splendid personalities the whole way across the world and improve your programming or DSA abilities as well as different abilities like legitimate and scientific reasoning, critical thinking, using time effectively, breaking an issue into little pieces, and numerous others.

Presently, this is the sort of thing that a significant number of them would've definitely known notwithstanding not being a competitive software engineer - however the issue with most people, particularly undergrads or fledgling developers, is they don't have the foggiest idea about the right and compelling way of beginning with Competitive Programming. Along these lines, with this application school or separate association's organization can without much of a stretch screen the understudies, guide the understudies, actually taking a look at every understudies execution very quickly. Generally assuming this works are done physically like checking and observing every one of the accessible entrances entering the data that would require long periods of long periods of time for every individual cluster then on the off chance that taken of entire school, it would take more time to finish such work. Yet, with utilizing this stage total comput-erization is accomplished and the work that is taking days to finish is currently will be done in the matter of seconds.

These are a portion of the users on which we have tried the calculation and there isn't a lot of distinction in genuine and anticipated values. This shows the exactness of the application that is sent. Furthermore, as I referenced already in backend algorithm that forecast of these values is profoundly inconceivable as it relies upon different variables that can know once challenge is finished. In any case, anticipating beforehand is inconceivable yet our calculation will provide users with an assessment of their advancement for the following half year.

Username	Actual Rating	Predicted Rating
sahithi 20	-88	-101
teja349	15	1
kp 1104	1	4
hardik24k9	-40	-32
priyanshi garg	22	29
vipresh2000	-8	-8
abhimanyu 17	-75	-64
taran910	19	12
yatinagg	19	11
siddhatgarg11	21	12
vsoni101	-28	-19
mscharan14	-58	-49
tiwari0000	-7	2
psyche37	-24	-31
pramod billa	-60	-51

Above in Fig 3[b] you can notice the forecast of a user and this is how the frontend shows the data graphically. On the off chance that you can notice the client is accomplishing something useful in CodeChef so even following a half year his advancement will tumble down somewhat however not more. Yet, if eyewitness the code forces chart that is populated it shows the user execution will go down quickly in the approaching half year. This shows the user isn't zeroing in on code forces stage more that is the reason he is having low advancement. This perception will be finished by the organization group of his foundation and separate direction will be given to the possibility for his improvement.

6 Conclusions

The assumption results will be instrumental in cultivating the programming sharpness of students in a tweaked manner. Observing the factors that could block the progress of students would simplify it to beat the moves that one could look at due to those components. Aside from anticipating the execution of the understudy, we assembled all their diligent effort at only one single application. Aside from foreseeing the outcomes, the introduced work contains itemized investigation to assist instructors with more deeply studying of different elements that influence the exhibition of understudies. From our discoveries, we presume that more modest and more controlled conditions give a superior stage to survey understudy execution. Accordingly, Performance anticipating apparatuses, as portrayed above, can be carried out in a learning climate to assist understudies with performing better in their programming tasks.

As a group of people yet to come residents, to upgrade expertise among youth is the primary saying and with cutting-edge forecasts and authoritative criticisms will help the associations and understudies to improve their programming abilities. India's populace is 1.38 Billion at this point and consistently around 3 Million designer understudies graduates so assuming that we consolidate these for each of the 3 or 4 years of graduate understudies it would be about 1 crore understudies. Indeed, a parcel of understudies need industry necessity abilities and there are different explanations behind it, one of the primary explanations is legitimate checking of understudies that should be possible through this application. Likewise, this application probably won't take care of the total issue yet this would add a stage to their outcome in their excursion. In future, we will chip away at the calculation more to work on the calculation so that a few factors can be anticipated and utilizing that variable exactness can be expanded for those clients whose expectation was not precise

References

1. Kaushal, R., Singh, A.: Automated evaluation of programming assignments. In: International Conference on Engineering Education: Innovative Practices and Future Trends, AICERA, July 2012. ISBN NO: 978-81-921320-1-3
2. Sagar, M., Kaushal, A.G.R.: Performance prediction and behavioural analysis of student programming ability. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE (2016). ISBN NO: 978-1-5090-2030-0

3. Devasia, T., Vinushree, T.P., Hegde, V.: Prediction of students' performance using Educational Data Mining. In: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE). IEEE (2016). ISBN No: 978-1-4673-8595-4
4. Sivasakthi, M.: Classification and prediction based data mining algorithms to predict students' introductory programming performance. Published in 2017 International Conference on Inventive Computing and Informatics (ICICI). IEEE (2017). ISBN:978- 1-5386-4032-6,
5. Potluri, S., Mohammad, G.B., Nandan Mohanty, S., Vaishnavi, M., Sahaja, K.: Secure intelligent framework for VANET: cloud based transportation model. *Cloud Secur. Tech. Appl.*, 145–170 (2021). <https://doi.org/10.1515/9783110732573-008>
6. Gouse Baig Mohammad, S., Kumar, P.R.: Integrated machine learning model for an URL phishing detection. *Int. J. Grid Distrib. Comput.* **14**(1), 513–529 (2022)