Our project started with the problem statement “The objective of our team is to develop a software solution to predict the future jobs based on location, sector, package and eligibility. Big data analysis can be useful to collect and analyze the data from different job sites and predict the future requirement applying machine learning”. Our project initialized by thinking over the requirements needed by our customer that is Government of Uttarakhand, we started studying and focusing over the criteria that how we can implement and accomplish it to a successful execution.

To bring our work in action firstly, we needed an architectural diagram through we which we can precede, where we came up with the solution that is how a user uses the platform, selects the criteria’s i.e. location, sector, package, salary and job title, then the attributes will get accessed by algorithm. To make it a more efficient site, we even worked that the output can even be of single value that is even if the user chooses any one of the attributes he/she can get a satisfying output. To give it a accurate and précised output we plotted the attributes with vacancies and with this plotting we were able to get a good predicted result as output that is as per the graphs plotted, if the vacancy is greater than mean vacancy, then our vacancy is low i.e. its demand in future is less and apart from it an output is generated.

To begin our work, we divided our project in frontend and backend. For a better and efficient frontend we required a complete layout of the wireframe in which we started the execution of the project with the index page which the user will visit it first. Keeping users interaction with website we kept the design simple and handy. So, firstly the user needs to login or signup to our website in order to get the desired job prediction, for which we have provided them login and signup buttons in header section. After successful login the user will be redirected to the home page of the website, in which we will take the input from him through simple dropdowns. This dropdowns are like education, job title, sector, city and salary expectations. After filling this form the data will be sent to the prediction algorithm and the corresponding output will be shown.

We tried to keep the design of the website simple, easy and user friendly to handle for the end user. We have even provided the proper buttons and links so that the user can go back and forth easily.

We have used frontend technologies such as HTML5, CSS3 and Bootstrap for designing the web pages.

To tackle with the backend our first problem was to generate the data which was not provided to us, it was the biggest challenge for us to generate a data that replicates the real world scenario of jobs. So, we explored the data, for example baggle, but we were unable to find the data that fulfils the needs of this problem statement. Then we explored many websites with the help of tools such as Dataminer and created a dataset that achieved what the problem statement desires. To clean the dataset we used numpy and pandas libraries.

Then we have implemented various machine learning algorithms after which we have framed XGboost and Random forest with good score, so, we further narrow it down to RMSE values. We choose XGboost as our final algorithm due to its low RMSE values as the RMSE value gives better accuracy.

We have even executed XGboost algorithm on our dataset which gives various predictions and a good accuracy score along with the execution of the graphs between the predicted values with other attributes to find the variance.

When we were working on the prediction we were getting problem while accessing on the attributes known as Eligibility. So, we used Count Vectorizer technique which helped us to tackle that problem while working on prediction.

To implement all the backend activities we used the following technologies:-

1. Sklearn for feature attraction and algorithms.

2. Matplotlib notebook for plotting.

3. Numpy and Pandas for data framing.

4. For prototyping we have used Jupyter Notebook.

5. We are further working on flask for connectivity and security.

6. MySQL for the database.

While working on the algorithms we found that the string has to convert into numeric values as ML accept better accuracy to it. So, we converted it into labels by using Label Encoding.

Since one attribute name Eligibility has having an issue with it so we can’t use Label Encoding, as each data in that particular cell has to separate it, so we use Count Vectorizer technique, which segregates each cell. After that we got better results while working with the algorithms.

We have worked on different algorithms and find out the accuracy scores and RMSE values. For more classification we went to insights and checked the learning curves which give us better knowledge about with algorithm to apply in our project.

We also plotted the graphs of the current data after training the model with other attributes which will help for the user to know for better approach. Then, we worked on the prediction graph which gives better idea for the future jobs in respect with vacancy and as well as we worked on the connectivity through flask.

We have worked on the prediction graphs which will give better idea for the future jobs in respect with vacancy.

Then our work for connectivity started with the help of flask where we have connected our index, signup, login and other html pages with flask framework. In signup page all the credentials will save in MySQL database. We have use SQLAlchemy for connectivity with databases.

In login page credentials we fulfilled all the criteria which will redirect to the main page and if not, message will be flashed with wrong credentials, where the credentials will be extracted from MySQL database to check if it is correct.

We have developed our main page which will take various inputs which will be processed with the backend connectivity. In which we have also worked on the result page where we will show the predicted output with different functions.