1. Importing dataset in Jupyter from excel.
2. Deleted the ‘Id’ column because I don’t think it is relevant for analysis.
3. Built understanding of the data.
4. The values in dataset columns are in different language that were making it tougher to analyse. So, I mapped the hindi value with the English value with the same meaning. These English values were already present in most of these columns.
    - df['scholarshipAvailed'].replace({'हाँ':1,'नही':0,'नहीं':0,'Not availed':0})
    - df['hasAdhar'].replace({'हाँ':'Yes','नही':'No'})
    - df['literacy'].replace({'शिक्षित':'Literate','नही':'Illiterate','अशिक्षित':'Illiterate'})
    - df['hasEnrolledAdultLiteracy'].replace({'हाँ':'Yes','नही':'No','नहीं':'No','#NAME?':'No'})
    - df['eduType'].replace({'औपचारिक':'Formal','अनौपचारिक':'Informal','#NAME?':'Formal '})
    - df['eduInformal'].replace({'व्यस्क साक्षरता':'Adult literacy','अन्य':'Others','शिक्षा पूर्ण':'Finished Education'})
    - df['statusFormalEdu'].replace({'शिक्षा पूर्ण':'Finished Education','जारी है':'Continuing','--- चुनिए ---':np.nan})
    - df['typeOfSchool'].replace({'सरकारी':'Government','निजी':'Private'})
    - df['eduTransport'].replace({'पैदल चलकर':'Walking','सार्वजनिक परिवहन - बस':'Public Transport','खुद का ट्रांसपोर्ट 2-व्हीलर':'Private Transport', 'सार्वजनिक परिवहन - ऑटो / रिक्शा':'Public Transport','खुद की ट्रांसपोर्ट साइकिल':'Private Transport','Public transport - bus':'Public Transport'})

    - df['isHeadOfFamily'].replace({'हाँ':'Yes','नही':'No','नहीं':'No'})
    - df['genderHeadOfFamily'].replace({'महिला':'Female','पुरुष':'Male','अन्य वर्ग':'Others'})
    - df['hasPDS'].replace({'हाँ':'Yes','नही':'No','नहीं':'No'})
    - df['hasSECC'].replace({'हाँ':'Yes','नही':'No','नहीं':'No','नंबर नहीं पता':'Yes'})
    - df['hasAyushmanBharat'].replace({'हाँ':'Yes','नही':'No','नहीं':'No','नंबर नहीं पता':'Yes'})
    - df['hasLand'].replace({'हाँ':'Yes','नहीं':'No'})
    - df['reasonLandless'].replace({'--- Select ---':np.nan})
    - df['hasHealthCert'].replace({'हाँ':'Yes','नहीं':'No','नहीं/नंबर':'No'})
    - df['employmentType'].replace({'स्व नियोक्ता':'Self Employed','वेतन भोगी':'Salaried','पूरा समय':'Salaried'})
    - df['isWageEarner'].replace({'हाँ':'Yes','नही':'No'})
    - df['isSHGMember'].replace({'हाँ':'Yes','नही':'No'})
    - df['isCooperativeMember'].replace({'नही':'No'})
    - df['isFPOMember'].replace({'नही':'No'})
    - df['noGroup'].replace({'नही':'No','हाँ':'Yes'})
    - df['wageRecievedCOVID'].replace({'मजदूरी के लिए अनुरोध नहीं किया':'Did not request for wages','पूर्ण':'Full','अनुरोध किया लेकिन मना कर दिया':'No', 'आंशिक - राशि INR में':'Partial - amount in INR','---चुनिये---':np.nan})
    - df['liveLostCOVID'].replace({'नही':'No'})
    - df['disabledCOVID'].replace({'नही':'No'})
    - df['illnessCOVID'].replace({'नही':'No','हाँ':'Yes'})
    - df['injuryCOVID'].replace({'नही':'No','हाँ':'Yes'})
    - df['movablePropLostCOVID'].replace({'नही':'No','हाँ':'Yes'})
    - df['immovablePropLostCOVID'].replace({'नही':'No','हाँ':'Yes'})
    - df['interestedCertProgram'].replace({'YES':'Yes'})
    - df['vocationCategory'].replace({'पर्यटन और आतिथ्य':'Tourism'})

5. **Imputation of Null values:**
   - **Target Variable- “scholarshipAvailed”-** As it is cleared in problem statement that government wants to help people who wants to continue their education. So based on that people who have finished the education, Illiterate, Informal education (education that is not in proper format), and Dropouts would not get the sponsorship. So, I imputed all those columns with zero. Also, It is also given in Problem statement that volunteers went to around 100+ people to encourage them to get sponsorship. This column had only 52 filled values. Where are the rest 60-70 values. I presumed they are the people who are still studying so I filled the ‘Scholarship availed’ column with 0 where education status is ‘continuing’ . According to me, there were no harm in filling those values with zero If I will not do that then I will have to delete these rows and the dataset is already an imbalanced one. So, I impute these values with 0 as well. In the end, only 47 nan values remained in dataset . I deleted those rows which are carrying NaN in ‘Scholarship availed’ column.

   - “literacy” and “Pincode” imputed with mode and median respectively.

   - Imputed “hasEnrolledAdultLiteracy” with ‘No’ because only illiterate people opted this and the missing values in this columns were all literate that means they haven’t opted this so ‘No’.

   - In ‘edutype’ imputed rows that had ‘literate’ in literacy column with ‘formal’ rest with ‘No_education’.

   - “eduInformal”- Impute it with mode where “edutype” is informal. Rest with ‘not taken’.

   - “statusFormalEdu”- Replace it with mode where edutype is ‘literate’. Mode is ‘Finished Education’. Rest with ‘No education’ because their ‘edutype’ was ‘Illiterate’.

   - ‘typeofschool’- Where edutype is ‘Literate’, Impute the null in ‘typeofschool’ with other values in column according to their probabilities in normalized column. Imputed rest null values with ‘No School’ because their edutype were ‘Illiterate’.

   - “eduTransport”- Replace null with No School were typeoschool is ‘No School’. Rest values as per their probabilities in normalized column.

   - “isHeadOfFamily”- Imputed it with mode i.e, ‘Yes’.

   - “relWithHeadOfFamily”- imputed with ‘Self’ because I found out that null is present in this column where “isHeadOfFamily” is ‘Yes’.

   - “genderHeadOfFamily”- imputed null values with other values present in column according to their probabilities in normalized column.

   - “hasSECC”- Imputed it with mode i.e, ‘No’.

- “hasLand”- Imputed with ‘No’ because if someone left that that means they don’t have or don’t know weather they have land or not which implies that they have no land.

- “reasonLandless”- Imputed it with ‘No hereditary ownership’ because I think if someone left the column blank that means he/she doesn’t know the reason of that means they have nothing to do with land. So,’ No hereditary ownership’.

- “employmentType”- Imputed with ‘Jobless’ because if someone left it that means he has no job.

- “isWageEarner”- Impute with ‘ No’ because all missing values had ‘Jobless’ in “employmentType” .

- “noGroup”- Imputed null values with other values(’Yes’ & ’No’) present in column according to their probabilities in normalized column.

- “isSHGMember”- Impute with ‘No’ where “noGroup” is ‘Yes’. Rest with other values(’Yes’ & ’No’) present in column according to their probabilities in normalized column where “noGroup” is ‘No’.

- “interestedCertProgram”- Impute with ‘No’ because if it is null that means that person is not interested in program regardless of their edutype.

- “wageRecievedCOVID”- Impute with ’Not eligible for wages’ where “employmentType” is’Jobless. In rest, Imputed null values with other values presented in column according to their probabilities in normalized column.

6. **New Feature Creation-**
    - Created a new column “Gender” according to the values present in “ relWithHeadOfFamily”. I created two list from all values of “relWithHeadOfFamily “. One with all female relation and second with all male relations and imputed ‘male’ where “relWithHeadOfFamily” values are from from list2 and ‘female’ where list1. Rest imputed with their value probabilities in normalized column.

7. **Droping of Tables-**

    - Dropped ‘pincode’ because I couldn’t use it as continous variable or if tried to convert it in categorical then it will create around 200 dummy variables. Also, after creating dummy I would not be able to use new ‘pincode’ values in future because after generation of model adding entire new column would be irrelevant. Deciding sponsorship based on ‘pincode’ would make model biased towards that particular ‘pincode’. It’s a ‘Pan Nation’ model not a single district.

    - Dropped ‘’relWithHeadOfFamily” because its variance has also been captured in new feature “Gender”.

- Also dropped column “ eduOther”,” isCooperativeMember”,” isFPOMember”, ” liveLostCOVID”, “disabledCOVID”, “illnessCOVID”, “injuryCOVID”, “movablePropLostCOVID”, “immovablePropLostCOVID” and “vocationCategory” because all of them have more than 90 percent missing values. Also, the values that are present in those columns are single values or second value in negligible amount so if I would try to impute then all values in one column would have same values so there will be zero variance. If there is no variance then it is not suitable for model building because you will not be able to pull out any information from it.

- Deleted “noScholarshipReason” and “eduOther” because their variance has been captured by other variables.

- Deleted 13 columns in total.

8. Outlier Removal- Deleted outliers from ‘Age’ column. Considered age between 5 and 80.

9. Dummy Variable Creation-

- popped “scholarshipAvailed” from dataset and push it into new series ‘y’ before creating dummy variables.
- Created dummy variables from all categorical variables and put these into new dataframe “dummy1”.
- New dataset “dummy1” now have 60 variables Including ‘age’ which is a continuous variable.

10. Splitted the dataframe into Train and Test(for both ‘dummy1’ and ‘y’).

11. Done Scalling (fit_transform) of variable ‘age’ in train and only transform in test.

12. Model Building-

- This Dataset has imbalance.

```
0     11034
 1       24
```

- **First model**- **<u>Logistic Regression + Class Weight (without any balancing)</u>**

  Accuracy = 0.998
  Precision = 1.0
  Recall = 0.375
  F1 Score = 0.545

  Recall is very low.

- **Second model**- **Logistic Regression + Class Weight (with class weight =’Balanced’)**

  Accuracy = 0.995
  Precision = 0.380
  Recall = 1.0
  F1 Score = 0.551

  Precision is low and F1 is also low.

- **Third model**- **Logistic Regression + Class Weight (with Grid Search for finding best weight)**

  Accuracy = 0.998

  Precision = 0.8

  Recall = 0.5

  F1 Score = 0.615

  Recall is low.

- **Fourth model- SMOTE + RandomForestClassifier**

  Accuracy = 0.998
  Precision = 0.83
  Recall = 0.62
  F1 Score = 0.714

  High Precision, F1 and Recall are fine too. Best model so far.

- **Fith model- Hyper Parameter Tuning in RandomForestClassifier (Using Grid Search CV)**
-

  Accuracy = 0.998

  Precision = 0.833

  Recall = 0.625

  F1 Score = 0.714

  High Precision, F1 and Recall are fine too. Same as fourth model.

- **<u>Sixth model- Smote+Logistic Regression</u>**

  Accuracy = 0.997

  Precision = 0.461

  Recall = 0.75

  F1 Score = 0.571

  Precision is relatively low.

- **<u>Logistic Regression Hypertuning (with Grid Search)</u>**

  Accuracy = 0.997
  Precision = 0.8
  Recall = 0.500
  F1 Score = 0.615

  Low recall.

13. I decided to predict probabilities of target variables with **fourth model.**

- Used predict_proba to predict values. It gave probabilities of both 0 and 1 so I took the second part i.e the probability of getting 1 by slicing and multiply by 100 for easiness.
- Created a dataframe with actual test values and predicted probabilities of 1.
- Created different probabilities cut-offs using for loop in same dataframe.
- Add a new variable to this dataframe that saves final label of each point. This will depend on the threshold probability taken by us. These label will change according to the threshold probability value.
- As mentioned in the Problem statement, we have taken seasonable threshold of 50.
- The model report is as follows.

  Accuracy = 0.999

  Precision = 0.714

  Recall = 1.0

  F1 Score = 0.833

- The cutoff probability is 50(0.50 in actual I multiplied problity with 100 for better calculation) which is fine because according to problem statement the cut off should be reasonable and it’s a welfare scheme so government would definitely want to award scholarship to more people.

- This model should capture as much ‘Yes’ as it can but also keep in mind that precision should be as high a possible because you don’t take a risk of finding more false positives because it would affect government spending. There should be a trade off between Precision and Recall. We can manage with reasonable Recall but Precision must be as much high as possible. Model 4 gives good Precision and decent Recall and F1 score so I think this model can be deployed.

14. Deployment Instructions- using Flask and Fast API

- I have used dropdown for taking the values in all variables accept age.
- Due to time constraint, I have not considered hindi values in dropdown but If I did these would be handled in the same way.
- First created a html(**labour.html**) template that takes the value of all variable which are used or creating final model. At the bottom of this file I have created a basic javascript function that would form a url in specific format on click on submit button. Used ‘Get’ method.
- Then created pickle file of **model4** (with cutoff probability of 0.5) from jupyter.
- After creating labour.html I created **api.py** that contains two routes one default and another one is predict. Default have a render function which show the labour.html template when someone run the api file from anaconda.
- In predict, Request module is used to get the different variables from url that’s been created after clicking on submit button. I used request.args method for that.
- Imported Joblib and used it to load pickle file(model_exp.pkl).
- Saved variable that I got from URL with help of request in new variables.
- Then I applied If – else for getting the right value for variables.
- Used predict function to predict and return the final output taking the saved variables as parameters.