****

**WiDS ‘22 - ‘23 Final Documentation**

**<Project UID - Name>**

**<Mentors>**

|  |  |  |
| --- | --- | --- |
| **Team Member Name** | **Roll Number** | **Email-Id** |
| G.Chanikya Prakash | 210050053 | chanikyaprakash2003@gmail.com |

**Introduction to Problem Statement**

|  |
| --- |
| 1. Basically, the problem statement is about how to detect/identify fraud cases from a bunch of transactions.  2. As you know, #fraud cases are far more lesser than #genuine cases  3. The key aspect is about “data balancing” |

**Existing Resources**

|  |
| --- |
| 1. https://www.youtube.com/watch?v=CmorAWRsCAw&list=PLeo1K3hjS3uuASpe-1LjfG5f14Bnozjwy  2. https://www.youtube.com/watch?v=3Xc3CA655Y4&ab\_channel=freeCodeCamp.org  3. https://www.youtube.com/watch?v=gmvvaobm7eQ&list=PLeo1K3hjS3uvCeTYTeyfe0-rN5r8zn9rw&index=1&t=0s  4. https://www.geeksforgeeks.org/machine-learning/  5. https://www.youtube.com/watch?v=JnlM4yLFNuo&ab\_channel=codebasics |

**Proposed Solution**

|  |
| --- |
| 1. First of all, I have cleaned the data and have checked all the transactions for any not-related/NA values.  2. We can try many methods to balance the data, I felt that using SMOTE gives the best results.  3. So, I have used SMOTE to create a balanced data.  4. Once the data is balanced, we can find the proper model to fit the data.  5. I found that, |

**Methodology & Progress (Mention the work done week-wise)**

|  |
| --- |
| Week-1:  Brush-up basics of python, pandas & matplotlib library  I have learned many new features/functions of pandas library from the tutorials  Week-2:  I have done the analysis of the data by drawing the relations b/w the parameters  Cleaned the data by checking for missing values and irrelevant values  Balanced the data using the SMOTE  Week-3:  Tried out different models while optimizing the score  Week-4:  Clearing the doubts with the mentor and making the report |

**Results**

|  |
| --- |
| 1. Number of fraud cases are negligible when compared to genuine cases  2. Model used by examining the different models – Randomforest  3. Data balancing algorithm used – SMOTE  4. By using PCA, we know that only 9 or 10 V\_i’s contribute to the result  5. Score is around 0.9999 for Randomforest by using 40 n\_estimators |

**Learning Value**

|  |
| --- |
| 1. I have learnt how to clean the data and analyse the data (EDA) using various features in Matplotlib  2. Also, I have learned how to use SMOTE to create a balanced data  3. I have learnt how to find a efficient model to fit the data |

**Tech-stack Used**

|  |
| --- |
| 1. numpy  2. matplotlib  3. sklearn  4. imblearn |

**Suggestions for others**

|  |
| --- |
| Nothing really.. |

**Contribution by each Team Member**

|  |
| --- |
| Basically, it is a individual project, so I have done the whole work :) |

**References and Citations**

|  |
| --- |
| Just used the references provided in the resources |