**Capstone Project Submission**

**Instructions:**

i) Please fill in all the required information.

ii) Avoid grammatical errors.

|  |
| --- |
| **Team Member’s Name, Email and Contribution:** |
| **Name:** Manisha Dhanuka  **Email Id:** [manishadhanuka5135@gmail.com](mailto:manishadhanuka5135@gmail.com)  **Contribution:**   1. Data Cleaning    * Handling Null values    * Duplicates Consideration 2. Data Exploration- Univariate Analysis:    * Qualitative measures of Data i.e., Descriptive Statistics    * Android apps distribution per Category    * Ratings Analysis with respect to app type: benign or malware    * Apps with the number of ratings given 3. Bivariate Analysis and Relationships with Label:  * Number of Dangerous permissions and Safe permissions with class labels. * Number of ratings and the ratings with class labels.  1. Data Preprocessing:  * Checking for Class Imbalance * Feature Selection * Categorical Variables Encoding  1. Data Modeling:  * Random Forest * Gradient Boosting * Support Vector Machine * K-Nearest Neighbors * Naïve Bayes  1. Data Validation and Evaluation:  * Recall Score * Precision Score * Accuracy * ROC-AUC Score  1. Hyper-parameter Tuning:  * Grid Search CV |
| **Please paste the GitHub Repo link.** |
| GitHub Link: - |
| **Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions. (200-400 words)** |
| Android is the most in demand operating system in mobile manufacturers community. But the problem of malware apps is a concern for all the users as well. In the project I worked with the apps related dataset to classify the apps as malware or benign.  First did the data cleaning by filling the null values and dropping the duplicates. Then explored the features and their relations with the labels.  Then, create the copy with only permissions indicator features, the category type and permission counts. Since, the category was a categorical column, used One-hot encoding to convert it into numerical feature.    For Data modelling, first did the train test split, after that first used tree-based methods as they don’t require scaling like boosting methods: Random-Forest, Gradient Boosting. After that scaled the data using Standard Scaler and applied K-Nearest neighbors, Support Vector Machines and Naïve Bayes and stored the Classification metrics in a dataframe and compared. After comparison we find that Gaussian Naïve Bayes was not working well. We did hyperparameter tuning on the KNN, Random Forest and Gradient Boosting since they were doing quite well in all the metrics. After comparing the scores, I decided to go and create a final model with Gradient boosting and then fit the entire data to the final model. |