Team Member's Name, Email and Contribution:

- Vinayak Marathe: vinmarathe100@live.com
- Riya Patel: riyapatelrp8308@gmail.com

Contributor's Role :

Vinayak Marathe :

- Worked on Problem Statement with Business Objective
- Datasets Reading and Data Wrangling
- Worked on Missing/Null values
- Prepared different type Model and implemented it (Logistic Regression, Decision Tree, Random Forest, Naïve Bayes)
- Performed Hypothesis Testing with 3 hypothetical statement
- Evaluated Model Performance/ Evaluation Matrices
- Worked on imbalanced class labels
- Worked on Competitive Advantages and Business Goals of our Project
- Presentation Preparation

Riya Patel :

- Data Explanation and Data Pre-processing
- Data Visualization and Storytelling
- Feature Manipulation and Feature Selection
- Prepared different type Model and implemented it (Gradient Boosting, XGBoost, K-Nearest Neighbor, Support Vector Machine)
- Performed Cross Validation and Hyper-Parameter Tuning
- Performed Model Explainability and Feature Importance
- Analyzed Overall Results of the Project /Insights
- Future Work & Conclusion
- Summary and Technical Documentation.

GitHub Repo link.

1. Vinayak Marathe: https://github.com/v1git12
2. Riya Patel: https://github.com/riyapatelrp

Project Link:
https://colab.research.google.com/drive/1x_MIL2gtjlIy__OYUtBzTNasteC8AST2?usp=sharing

Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions. (200-400 words)

The project Android Authenticity Prediction (AAP) system uses machine learning techniques to classify applications as authentic or malicious based on a set of application features extracted from static analysis of the application code. The first Android smartphone was launched in September 2008, and shortly thereafter, smartphones powered by the new open-source operating system were everywhere. In 2021, almost 12 new enhanced versions of Android were released, and it is the most widely used mobile operating system in the world, with an 84% share of the global smartphone market. With this level of adoption coupled with the open-source nature of Android applications, security attacks are becoming more and more ubiquitous and seriously threaten the integrity of Android applications. Statistics show that more than 50 million malware and potentially unwanted applications have been identified for Android.

The problem of Android Authenticity Prediction is to develop a machine learning model that can accurately predict whether an Android application (app) is authentic or not. With the increase in the number of mobile apps, the risk of downloading malicious apps has also increased. Malicious apps can steal sensitive user data, perform unwanted actions, and damage the user's device. Therefore, it is essential to develop a model that can accurately predict the authenticity of Android apps and help users make informed decisions about which apps to download and install on their devices. The challenge is to identify the relevant features that can distinguish between authentic and malicious apps and to train a classification model that can generalize well to new, unseen apps. Additionally, the model should be able to handle the large and dynamic nature of the mobile app ecosystem, where new apps are constantly being developed and released.

The proposed approach's implementation begins with downloading the dataset. Then data wrangling and feature manipulation is executed as a step of pre-processing of data where we get to know that our target variable i.e. class label is imbalanced so we handle class label imbalance using different types of techniques. After this, the data is analyzed and a different model is executed. Then we have done the hypothesis testing with 3 different hypothetical statements. And at the next stage we have implemented different types of classification techniques and performed cross validation & hyper parameter tuning to find the best optimal hyper parameters for better performance. At last, all the business insights carried out in this project. Let's know the step-wise approach of this project.

Our first task is to prepare a dataset for our machine learning model. First we load the dataset, we start with the **Know Your Data** it involves Null Values/Missing Values, Unique Values, Duplicate Values. And we found that some of the features in our dataset have a missing value so we have to deal with it as well as with duplicate values. Then the **Data Wrangling** comes, in this we make our dataset analysis ready and deep dive into the relationships of variables by knowing the value counts and unique categories of the mobile application and find some important insights regarding the ratings of the apps and reviews.

Next, in the **Data Visualization**, figuring out various aspects and relationships among the target and the independent variables. We have done certain steps to know the relation by using different Charts and Graphs (using python libraries seaborn, matplotlib) like Bar plot, Count Plot, Pie Chart, Hist Plot, Correlation Heat map, pair plot etc. Here, we found the insights regarding the categories of the application, rating of the application, number of reviews and most important price of the application. We also found here that our dataset has an imbalanced class label so we have to work on it.

Then the next step is **Hypothesis Testing**, in which we define three statements from the dataset to obtain a final conclusion about the statements through code and statistical testing. First hypothetical statement is: is The app's rating a significant predictor of its authenticity? Second, is The app's category a significant predictor of its authenticity? Then the third and last statement is The number of reviews a significant predictor of an app's authenticity? In these three hypothetical statements, we research the hypothesis as a null hypothesis and alternate hypothesis then perform an appropriate statistical test.

Next approach is the **Feature Engineering & Data Pre-process**ing, in this we handle the missing values of some columns, Encoded our Categorical variable, Perform Textual Data Preprocessing where we removed punctuations and some other language's app name, then done Feature Manipulation & Selection where we have dropped some unimportant columns and added new columns using one-hot encoding, handled imbalanced data using resampling, tomek links & smote techniques and finally, splitted our Data into training and testing set using 80-20 ratio i.e. we divide our train and test data with 80:20 ratio.

Next comes the most important part of the project is the **ML Model Implementation**. Here, we also performed some cross validation and hyper-parameter tuning techniques for better and optimal hyper-parameters so that we can get the best parameter as well as the great accuracy. So, Firstly, we use Logistic Regression which gives us a good result of 70% accuracy. Then, we implemented a Decision Tree Classifier which gave us an accuracy of 77.32%. After this, we have used a Random Forest Classifier and found an accuracy score of 88.24% by using a smote technique which is quite good among all the techniques we have used so far. Then, we used Naïve Bayes and KNN classifiers in the search for more accurate results but these algorithms gave us very minimal results of 66.85% and 77.26%. So then we have tried the most popular and fast ensemble techniques named Gradient Boost and XGBoost where we achieved the accuracy score of 88.34% and 89.92% again by using smote technique which is again a great and highest accuracy as compared to naïve bayes and knn and any other models. It is nearest to the accuracy of random forest technique. And lastly we performed Support Vector Machine i.e. SVM classifier using cross validation and tuning and this gave us a bad accuracy of 66.92%.

In the end, we found that the most accurate and optimal algorithm is XGBoost followed by Random Forest and Gradient Boost.

At the last stage of our Android Authenticity Prediction classification project, we found the model which we have used and found the feature importance using shap model explainability tool to explain predictions. The feature importance analysis has identified that the **number of ratings** is the most important feature followed by rating and price. This indicates that these features play a significant role in predicting the authenticity of an Android app.

In conclusion, our goal was to use classification techniques to solve the problem of detecting malware applications on Android phones. We experimented with different supervised classification techniques and identified the best technique for each approach. The results obtained from this project show that artificial intelligence can be used effectively to detect and prevent malicious Android app downloads. This authenticator can save Android users from the potential dangers of downloading and installing malicious Android apps. By incorporating this AI-based authenticator into existing anti-malware safeguards, users can rest assured that their devices are protected from a wide variety of malicious threats. In addition, Random Forest, XGBoost and Gradient Boosting Classifiers provide great results in this approach.