submission files:

1. ready to use code file in jupyter notebook
2. report in PDF
3. CSV file containing the classification predictions, including the probabilities. Please refer to the example file “result_example.csv” for the column names

Introduction:

In this project, you will be given a set of features about executable files (.exe), and your task is to determine whether the file is malicious or benign. This will be done through static analysis of the files, which means analyzing the file information without executing them.

The project focuses on Binary Classification, where you are required to classify records into two categories: whether the file is malicious (1) or not (0), based on a set of features in the dataset. Some of the features are known, while others are anonymous.

The team does not intend to limit you in terms of thinking and work approach, but there are some basic guidelines that you should follow.

General Guidelines:

- Do not prepare or design the raw dataset using Excel. Use Python packages instead.

- Implement the code in Jupyter Notebook within the Anaconda environment. Include full explanations in the code itself using markdowns and comments.

- You are allowed to use all the packages that come with the Anaconda environment. If you need to use additional packages that require installation, obtain approval from the course teaching assistant. If approved, include a dedicated cell in the notebook for installing the additional packages using "!pip install...".

- Avoid displaying warnings in the final notebook by adding the following code:

  ```python

  import warnings

  warnings.filterwarnings('ignore')
  ```

- The runtime of the code from start to finish should not exceed one hour.

- Do not use external files except for "csv.test" and "csv.train." The "test" file does not contain labels.

- As you will learn during the semester, there is no obligation to use all the features in the dataset. You can engineer and design the features as you see fit.

- The performance of your model will be evaluated using the AUC metric.

- The use of functions and techniques not covered in the course is welcome, but they should not replace traditional methods.

- Bonus points will be awarded based on the performance of your model: 7 points for first place, 4 points for second place, and 1 point for third place.

- A significant penalty will be applied for code that does not run and for a CSV file that is not submitted in the requested format.

- Specify the assumptions made at each stage of the project. Unspecified assumptions will be considered as if they were not taken into account, which may result in a lower grade.

- For each visualization you present, provide an explanation of the insights gained from it or the interesting aspect it represents. A visualization without an explanation will be considered nonexistent.

- Even if you attempted to approach the problem in a certain way and did not achieve any improvement in the results, do not remove the attempt from the code. Simply emphasize that it was an unsuccessful attempt and does not belong in the final workflow.

- The notebook should tell a story - how you explored the data and improved your model by trying different approaches (even if they failed).

Programming Task

Part 1 - Exploration:

● You need to explore the data in every way that comes to mind: the distribution of each feature, correlation behavior between features, statistical data on the features. At this stage of the project, there is plenty of room for visualization! Take advantage of it. The main focus will be on the conclusions drawn from this stage.

Part 2 - Preliminary Processing:

For the questions appearing in the sections, you should answer within the notebook's body (Markdown) alongside the relevant code snippets.

● Are there any outliers in the data? If yes, you should remove them or at least discuss them.

● Are the data normalized? If not, should you normalize them? What is the importance of normalizing the data in the problem?

● Are there any missing data? How did you choose to handle them and why in this way?

● Dealing with categorical variables.

● Is the dimensionality of the problem too large? Why can a large dimensionality create a problem? How can you identify that the dimensionality of the problem is too large?

● Reducing dimensionality using a technique learned in class - PCA, and/or by selecting a subset of existing features (feature selection). How did reducing dimensionality affect the model?

● Creating new features and/or mathematical manipulation of existing features

● Applying the preliminary processing on the Test set

● Additional experiments can be performed that were not taught in the course in order to process the given features (bonus).

Part 3 - Model Execution:

- Building two initial models out of the following three and applying them to the dataset:

  - Naïve Bayes Classifier
  - KNN
  - Logistic Regression

- Choosing two advanced models out of the following four and applying them to the dataset:

  - Multi-Layer Perceptron (ANN)
  - Decision Tree
  - Random Forest or Adaptive Boosting
  - Support Vector Machines

- You should explain the significance of the hyperparameters you chose to change and how they affect the model in terms of diversity and bias (appendices may be included).

- Try to explain the contribution (importance) of each feature to the success of the model as much as possible (bonus points will be awarded for impressive analysis in this regard).

Part Four - Model Evaluation:

- Building a Confusion Matrix on one of the models and explaining what the cells in the matrix mean in the context of the chosen model, i.e., what can be inferred about the model's performance in this context.

- Evaluating the model using Validation Cross Fold-K and constructing ROC plots for each Fold-K for each of the executed models (preferably in the same graph)

- Performance differences between running the model on Train or Validation data: is your model overfitted? What did you do or should you do to increase its generalization ability?

Part Five - Prediction Execution:

- After selecting the model, you need to make predictions on the "csv.test" file and save it in a "csv" file named "csv.number_group_results" (replace "number_group" with the group number), which includes the probability predictions (Probabilities Prediction - see attached example) for each of the observations in the "csv.test" file.

● Predictions should be made for each of the observations in the "csv.test" file.

● It is essential to create a pipeline at the end of the notebook for running the final model. This means a section in the notebook that includes the entire process from the beginning to the end (data loading, preliminary processing, and prediction) using the appropriate functions developed throughout the notebook.

Part Six - Use of Uncovered Tools:

• Describe at least one tool that you used in the project that was not covered in the course.

• You can choose any part of the project to apply this tool - exploration, preliminary processing, modeling, evaluation.

• Explain why you chose to use it, how it works, and how it affected the model.

Notes:

1. It is unnecessary to mention that the models should be evaluated on a validation set rather than the train set itself. Evaluating a model based on its performance on the train set may result in very low scores and significant point deductions. Running the evaluation on the train set can assist in identifying overfitting but does not serve as an indicator of the model's quality.

2. You should explicitly mention the hyperparameters of the chosen models, even if you decide to use their default values.

3. The order of implementing the stages is not mandatory. In a data science project, it is common to go back to earlier stages (similar to any science project).

Summary Report

The report will consist of a maximum of 5 pages (excluding cover and appendices), in which you will explain all the steps taken during the data analysis process. A detailed explanation is required regarding the response to the previously asked questions, the rationale behind the selection of each method mentioned in the preceding steps, the chosen hyperparameters, and the results of the different models. There is no need to elaborate unnecessarily or quote course materials in the project.

In a less formal aspect, this is the place to explain the overall process we performed, including dilemmas, decisions, and so on (the "story").