As part of my ML course I'm attending, we have a final project assignment.

I will now give you our course syllabus and then the full project details:

ML Syllabus

- Intro, and a mathematical recap (linear algebra etc.)
- The learning schema components, and the linear model
- The Bias-Variance trade-off
- Model selection and dimensionality reduction
- Model evaluation and Logistic Regression
- Deep Learning
- Support Vector Machines
- Kernels and Decision Trees
- Tree based algorithms
- Probability based Classification
- Probability distribution estimation
- Unsupervised learning: Clustering and anomaly detection
- TFIDF, selected advanced fields,

optional readings:

- Elements of Statistical Learning Theory, by Hastie, R. Tibshirani and J. Friedman, Springer 2009.
- Introduction to Statistical Learning, by Hastie, R. Tibshirani and J. Friedman, Springer 2009.
- Understanding Machine Learning: From Theory to Algorithms, by Shai Shalev-Shwatz, Shai BenDavid.
- Pattern Recognition and Machine Learning, by Christopher M. Bishop.

The project Assignment we were given:

Project Introduction

In this project you will be given several features about executable (exe) files. and your task is to decide if the file is malicious (malicious) or not (benign),

Through static analysis of the files - that is, analyzing the information about the files without running them.

In the project we will deal with the Binary Classification problem (that is - from two classes) in which you have to classify records into two categories - is the file malicious (1)

or not (0), based on a number of features in the data set. Some of the features are known and some are anonymous.

The goal of the final project is to expose you to practical work where you can experience the material taught in the course, while applying the tools in the environment

Absolutely real data.

The team does not intend to limit you in the way of thinking and working, but there are several basic guidelines that you must comply with.

General instructions

● Do not prepare and format the raw data set using Excel (but only in python packages).

● The implementation of the code will be done in a Jupiter notebook in the Anaconda environment, and will include full explanations of the nature of the implementation in the code itself (with the help of markdowns and comments within the code)

● It is allowed to use all the packages that come with the Anaconda environment. Use of additional packages that require installation is possible with the approval of the course instructor. If such approval is received, a dedicated compartment for installing the additional packages must be included in the notebook.

(!pip install…)

● Avoid displaying warnings in the final notebook - this can be done using:

```
import warnings

warnings.filterwarnings('ignore')
```

● The duration of running the code from start to finish should not exceed an hour.

● Do not use external files except for csv.test, csv.train when the test file does not have the labels.

● As you will learn during the semester, it is not mandatory to use all the features of the data set. You can engineer and design the features as you wish.

● The performance quality of your model will be determined by the AUC metric

● Uses of functions and techniques not taught in the course are welcome, but they are not a substitute for traditional methods.

● A bonus will be distributed to students according to the performance of their model - 7 points for first place, 4 points for second place, one point for third place.

● A large fine will be given for code that does not run and for a csv file that is not submitted in the requested format.

● The assumptions taken at each stage of the project must be specified. Discounts that are not specified will be considered as if they were not taken into account and this situation may lead to a lowering of the grade.

● For each visualization you present, you must write an explanation of what you deduced from it and/or what part of it is interesting. Visualization without explanation will be considered non-existent.

● Even if you tried to "crack" the problem in a certain way and there is no improvement in the results: do not remove the attempt from the code. It is only important that you emphasize that this is an unsuccessful attempt and has no place in the final work flow.

● The notebook should tell a story - how you studied the data and how you improved your model when you tried solutions from different directions (even if they failed).

The programming task (the score for the various sections in parentheses)

First part - exploration (8 points)

● You must explore the data in any way you can think of: the nature in which each feature is distributed, correlative behavior between the features, statistical data on the features. At this stage of the project there is a lot of room for visualization! Take advantage of it. A grade will be given mainly on the conclusions obtained from this stage.

Part two - pre-processing (35 points)

For the questions that appear in the sections, you must answer in the body of the notebook (Markdown) next to the relevant parts of the code

● Are there outliers in the data? If so, you should remove them or at least consider them (3)

● Is the data normalized? If not - should they be normalized? What is the importance of normalizing the data in the problem? (3)

● Are there any missing data? How did you choose to treat them and why in this way? (3)

● Dealing with categorical variables (4)

● Is the dimensionality of the problem too great? Why can large dimensionality create a problem? How will we recognize that the dimensions of the problem are too large? (6)

● Dimensionality reduction by one technique learned in class - PCA, and/or by selecting a subset of existing features

selection). How did the reduction in dimensionality affect the model? (6)

● Building new features and/or mathematical manipulation of existing features (2)

● Applying the preprocessing to the Test set (8)

● You can make additional attempts that were not taught in the course in order to process the features given to you (bonus)

Part three - running the models (20 points)

● Building two initial models from the following three and applying them to the data set: (10)

Naïve Bayes Classifier ▪

KNN ▪

Logistic Regression ▪

● Selecting two advanced models from the following four and applying them to the data set: (10)

Multi-Layer Perceptron (ANN) ▪

Decision Tree ▪

Random Forest or Adaptive Boosting ▪

Support Vectors Machine ▪

● An explanation must be given about the meaning of the hyper parameters you chose to change and how they affect the model in terms of variance and bias (can appear in the appendices).

● If possible, try to explain the contribution (importance) of each feature to the success of the model. (a bonus will be given for an impressive analysis in this aspect).

Fourth part - evaluation of the models (23 points)

● Constructing a Confusion Matrix (model) on one of the models, you must explain what the cells in the matrix say in relation to the model you have chosen, that is, what can be concluded about the performance of the model in this context. (10)

● Evaluating the model using Cross Fold-K Validation, and building a ROC output on each Fold-K for each of the models that were run (preferably on the same chart). (8)

● Performance gaps between running the model on the Train or on the Validation, is your model overfitted? What have you done / should you do in order to increase its ability to generalize? (5)

Fifth part - making a prediction (9 points)

● After choosing the model, you must make a prediction on the "csv.test" file data, and save it in the csv file.}number_group_{results

(Replace number_group with the group number) which includes the classification probability predictions (Probabilities Prediction - example).

● A forecast must be made on each of the observations appearing in the csv.test file

● Very important - you must create a pipeline at the end of the notebook to run the final model.

That is, a part of the notebook where there is the process from beginning to end (from loading the data and performing pre-processing to performing prediction) with the way you found it appropriate to use it for processing and prediction (the model). This part is intended for fast and reliable reproduction of results, where you will use the various functions you have developed throughout the notebook (5)

● Minimum Test AUC 0.9 : (4)

Part Six - Using tools not learned in the course (5 points)

• You must describe at least one tool that you implemented in the project and that we did not learn about in the course.

• You can choose any part of the project to implement this part - exploration, pre-processing, learning, evaluation.

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

Remarks:

1. It goes without saying that you will have to test the models on the Validation set and not on the Train itself (choosing a model according to its performance on the Train may lead to very low results and a significant lowering of the score!). Running the predictions on the Train can help in finding Overfitting but does not constitute An indicator of the nature of the model.

2. You must explicitly write the hyper parameters of the selected models as learned in class, even if it was decided to use their default values.

3. The order of implementation of the steps is not binding, within the project it is very likely that you will need to go back to earlier steps (similar to any Science Data project).

The final report

The report will include a maximum of 5 pages (not including the title page and appendices) in which all the steps taken during the data analysis will be explained. A detailed explanation is required of the answers to the questions asked above, the rationale behind the selection of each of the methods specified in the steps above, of the selected hyperparameters, and the results The different models. There is no need to expand in words and there is no need to quote the course material in the project.

In a less formal aspect, this is the place to explain the whole process you performed. Dilemmas, decisions, etc. ("story").

The report will be written in Hebrew or English in Calibri font, with a line spacing of 1.15

● An appendix must be attached to the report that includes an explanation of the responsibility of each partner and his contribution to the work. Omitting this part will lead to a significant lowering of the grade.

● The assumptions taken at each stage of the project must be specified. Discounts that are not specified will be considered as if they were not taken into account and this situation may lead to a lowering of the grade.

● You can add as many appendices as you like (and refer to them in the code). Visualization will appear in the appendices (and also in the body of the code)