**Machine Learning Course - Final Project**

In this project each team (3 members) will receive a different dataset and will present a 10 minute presentation at June 28th about the machine learning problem the team tried to solve. The presentation and notebook used to create it must be submitted by June 27th.

**Datasets:**

|  |  |
| --- | --- |
| **Dataset** | **Description** |
| Big mart sales | Predict product sales in supermarkets |
| Adult | Predict whether a person makes more than 50K a year |
| Default of credit card clients | Predict whether a client will default |
| Apps ratings | Predict an app rating |
| Online news popularity | Predict number of shares of an article |
| HR promotions | Predict whether an employee will be promoted |
| Food demand | Predict the number of orders of a meal |
| House prices | Predict the price of a house |
| Boston Airbnb | Predict the price of renting an airbnb |
| Hotel bookings | Predict whether a booking will be canceled |
| Heart | Predict whether a patient has a heart disease |
| Churn | Predict whether a customer will churn |
| Diabetes | Predict whether a patient has diabetes |

**Files:**

In addition to this document please find zip file “data and example.zip", the file contains:

1. **Datasets**– the folder includes all the above datasets
2. **Load Data.ipynb** – a notebook that uploads all datasets into a pandas data frame. Copy the code of your dataset and use it for your notebook.
3. **Titanic example** – a notebook + the titanic dataset csv file (before preprocessing) with examples on how to do this project.

**Teams registering for datasets:**

Using the “final project” forum. Please post, until June 9th, the team members, an email of one of the team members and the dataset you will be working on. Note that datasets already assigned to other teams (previous posts) are not available for you to choose.

**Presentation:**

The presentation should include the following aspects:

1. Names and IDs of all team members
2. The problem: what are you trying to predict? What data is available for that (input)? What can be the motivation and applications for solving the problem?
3. Evaluation: what metric(s) is used to evaluate the performance of an algorithm? How did you split the data for validating your model performance?
4. Data description:
   1. How many examples are in the dataset? (train/test)
   2. How many features are in the dataset?
   3. What is the distribution of the labels?
   4. Are there any missing values?
   5. Show 2-3 graphs describing various aspects of the data.
5. Data engineering:
   1. Did you remove any features?
   2. Did you add any features?
   3. What did you do with missing values?
6. Algorithms performance: Compare and analyze the train and test performance (according to the evaluation metric) of the following algorithms
   1. Benchmark algorithm:
      1. Regression benchmark – always predict the train set average label value (see example in the real and synthetic data regressions notebooks).
      2. Classification benchmark – always predict the train set majority class (see example in the final project example notebook).
   2. ML Algorithms: test the following algorithms:
      1. K nearest neighbors, try 3 different k values.
      2. K nearest neighbors with scaled values, try 3 different k values.
      3. Decision tree with 3 different max depth values.
      4. Random Forest with 3 different max depth values and 100 estimators.
      5. Ada Boost with 3 different max depth values and 100 estimators.
      6. Lasso regression with scaled values, try 3 different alpha values (only for regression)
7. Algorithms introspection – inspect the various algorithms artifacts. What decision trees did you get? What are the weights of the lasso coefficients? What is the random forest feature importance? (answer at least one of the questions listed).
8. Hyperparameters: using the best performing algorithm you found. Look for the hyper parameters of the algorithm that best improves the test performance, show a comparison in performance using their values.
9. Additional analysis (1 from the below):
   1. Performance vs. amount of data: Using the best performing algorithm. Show a graph describing the test performance of the algorithm when using [10%/30%/50%/70%/100%] of the train set for training the algorithm. Would you recommend collecting more data for the problem?
   2. Averaging models: Choose 2 or 3 different models tested in section 6. Now create a “new” model which predicts the average of the models you chose. Compare the performance of the new algorithm to all other algorithms you tested.