Project Milestone 3

Applied Data Science using Python
New York University, Abu Dhabi

|| Out: 14 Nov 2023 || Due: 5 Dec 2023 ||

The third project milestone will be in the form of an **in-class presentation**. Here are some high-level guidelines:

- Each **presentation should be limited to 15 minutes**. This is a strict time limit to accommodate all the groups.

- Presentations should include a **"walkthrough" of your Jupyter Notebook**, and if needed a small deck of powerpoint slides. We are most interested to see the code and techniques used by your group to explore and clean the data.

- All the **code should be written in Python**.

- **Jupyter notebook should be submitted to Brightspace**. Detailed explanation of your thought process in each step should be included in the text cells and as comments in the notebook. Your notebook should be comprehensive.

In the first 2 milestones of the project, we expected you to (a) prep your data (b) form your hypotheses and problem statement, and (c) perform some exploratory data analysis using visualizations and statistical analyses.

In this milestone, we expect you to **create your predictive model** in order to support your problem statement. You are expected to achieve this using the following three steps:

1. **Continue with your exploratory data analysis (EDA)**, if needed, based on the feedback provided to you in milestone 2.

   By completing milestone 2, you should have gotten a deeper understanding of your data's statistical characteristics, created visualizations, and tested your hypotheses.

   To summarize, there are four main types of EDA:

   - **Univariate non-graphical:** allows you to make observations of the population and understand sample distribution of a single variable, for example:
     - The *measure of spread* which describes how similar or varied the set of observed values are for a particular variable (data item). Measures of spread include the range, quartiles and the interquartile range, variance and standard deviation.
     - The *measure of central tendency*, which helps you find the middle, or the average, of a dataset. The 3 most common measures of central tendency are the mode, median, and mean.
     - *Outlier detection*

- **Univariate graphical:** graphical analysis on a single variable, for example:
    - Histograms
    - Box Plots
    - Violin plots
- **Multivariate non-graphical:** techniques which show the relationship between two or more variables. For example,
    - Correlations
    - Covariance
- **Multivariate graphical:** graphically show the relationship between two or more variables, for example:
    - Bar plots
    - Scatterplots

Remember, the aim of EDA is to find underlying patterns within the data, detect outliers and test assumptions with the final aim of finding a model that fits the data well.

2. **Perform some feature engineering**, in order to identify your most informative features.

Remember, a "feature" is an attribute of a dataset that is useful to the problem statement you are trying to solve. If a feature has no impact on the problem you are solving, it should not be part of the problem.

So what is feature engineering? Feature engineering is defined as the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.

The better the features you create and choose for your predictive model, the better your results will be. Feature engineering is an "art" where you decompose or aggregate raw data to help solve your problem. There are, however, multiple approaches to this process.

- **Feature Selection:** selecting the features which contribute most to the problem you are solving. For example using pearson correlation or variance thresholding.
- **Feature Construction:** the process of manually building more efficient features from raw data.
- **Feature Learning:** the automatic identification and use of features, eg: using K-means clustering (unsupervised learning) to identify new groups.

Using feature importance scoring methods, you can estimate how useful the feature will be. Features are given scores so they can be ranked based on these scores. Such methods include:

- **The correlation coefficient** between the feature and the target variable, ie the feature you are trying to predict.
- **Co-integration between two time series**, in the case of time-series data
- **Some predictive models have embedded feature selection methods**, so you don't have to perform any feature selection e.g. Random Forest
- **Chi-Square test** between target and numerical variables.

Note that it is normal to find yourself returning to this step multiple times. Feature engineering is an iterative process. It can look something like this:

1. Brainstorm feature ideas
2. Create features based on the problem statement
3. Choose features based on feature importance scores
4. Calculate model accuracy using the chosen features on unseen data
5. Repeat the steps until a suitable model is chosen.

3. **Create a predictive model** to support your problem statement.

As we learned in class, all machine learning models are classified either as *supervised or unsupervised learning* problems. A supervised problem is where a function maps an input to an output (label) based on input-output pairs. The machine learning model learns from the input-output training data to make predictions on unseen data (test data). Supervised learning problems are labelled as a **Regression** (output variable is a real value) or **Classification** (output variable is categorical) problem. An unsupervised problem is where a model looks for patterns within an unlabelled dataset.

Now creating your predictive model can be done as follows:

1. Preprocessing

Data preprocessing helps to enhance your data quality by organizing raw data in a suitable format to build and train a machine learning model. You are expected to:

1. **Standardize or normalize your data** if the model's algorithm is sensitive to unscaled data.
2. **Split your data into train and test datasets**. This is important as you don't want to contaminate the training data with the test data. You also need to split them into target variable (what you are trying to predict) and predictor variables (the features you are using to predict the target variable).

2. Select your machine learning model

As previously mentioned, machine learning models are classified as supervised or unsupervised. Your task is to select the best model to fit your problem statement. Here are some models outlines based on their categories for you to explore:

1. **Supervised - Regression**: Linear Regression, Multivariate Linear Regression, Support Vector Regression (SVR), Random Forest
2. **Supervised - Classification:** k-Nearest Neighbor (kNN), Logistic Regression, Support Vector Machines (SVM), Random Forest
3. **Unsupervised:** K-means clustering, Principal Component Analysis (PCA)

After building a few machine learning models, the models need to be trained by tuning the hyperparameters to optimize the model's performance. By comparing the predefined metrics for each model, an optimal model can be chosen. Below are some metrics you can use to compare your models' accuracy:

- **Regression metrics:** Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-Squared, Adjusted R-Squared.
- **Classification metrics:** Accuracy, Precision, Recall.

Grading

This milestone constitutes **5%** of your final grade, and will be graded from 100 points. All steps and work must be shown. The breakdown/rubric is as follows:

Category	**Points**
Thorough EDA including both graphical and non-graphical elements	10
Well thought feature engineering	10
Justified selection of informative features to feed into model using feature importance scoring methods,	15
Preprocessed data and correct selection of machine learning model with clear justification for model selection	15
Train model and tune its hyperparameters to optimize the model's performance. Select and show appropriate model accuracies to show optimal model performance.	30
Presentation	10
Well explained notebook with all the relevant code, analysis and visualizations	10