1. **Setting up an Integrated Development Environment (IDE)**

   There are many ways to setup the IDE. However, we recommended using Anaconda, which is a pre-built python environment with bundles of useful packages.

   (a) Download Anaconda from: `https://www.anaconda.com/distribution/`. Chooose the correct version based on your operating system and install it step by step.

   (b) Configure the PATH environment variable to make the conda command work. The following command is an easy way for testing whether your configuration is correct. If it is, you will see the version of conda being outputed. Otherwise, for macOS or Linux, the /path/to/anaconda/bin should be appended to PATH, and for Windows, /path/to/anaconda, /path/to/anaconda/Library/mingw-w64/bin, /path/to/anaconda/Library/usr/bin, /path/to/anaconda/Library/bin, /path/to/anaconda/Scripts should be appended to PATH. [1]

   command: conda –version
   sample output: conda 4.6.12

2. **Google Colab**

   Google Colab is a great resource for any Machine Learning class. Follow the steps in `https://colab.research.google.com/notebooks/intro.ipynb` to become more familiar with Google Colab.

3. **Pandas**

   This section is for practicing basic funtions of the Pandas library using the Salaries data set.

   (a) Consider the Salaries.csv file.

   (b) Use the read_csv(...) method from Pandas (**Documentation Link**) to read data from file Salaries.csv and to copy it into a dataframe.

   (c) Make the column playerID in the csv file as the index column and the first row as the header. Also, skip the second row when reading the file.

   (d) Select the id of the players who are registered in ATL and HOU whose salary is higher than one million.

   (e) Use the describe() method to calculate the standard deviation, first quartile, median, third quartile, mean, maximum, and minimum of the salary in team ATL.

   (f) Create a Python dictionary object whose keys are the headers of the dataframe created in the read_csv() exercise and values are Python list objects that contain data corresponding to the headers. (Here, use the iterrows() method to iterate each row of the dataframe and copy it to a dictionary. However, there is an easier way. Learn how the to_dict() method works by yourself later)

[1] If you have no idea with what the PATH environment variable is, please use the resources on the Internet to learn about it.

(g) Create a dataframe using pd.DataFrameRead (**Documentation Link**) and from the dictionary created in (e). Then, change the header to "a", "b", "c", ...

4. **Numpy**

Quick start:
https://www.numpy.org/devdocs/user/quickstart.html

Numpy axes explaination:
https://www.sharpsightlabs.com/blog/numpy-axes-explained/

(a) Create a 2-dimensional Python list object, then convert it to a Numpy array object.

(b) Examine the ndim, shape, size, dtype, itemsize, and data attributes of the numpy array object. Make sure you understand their functions.

(c) Learn the dimension concept of an ndarray object by using reshape() and flatten() methods.

(d) Understand how the slice operation works for 1-D arrays and 2-D arrays and practice by yourself.

(e) Learn operations on ndarray by examining the argmin(), argmax(), min(), max(), mean(), sum(), std(), dot(), square(), sqrt(), abs(). exp(), sign(), and mod() methods. Make yourself comfortable with these methods.

(f) Examine the arange(), ones(), zeros(), eye(), linspace(), and concatenate() methods. Make yourself comfortable with these methods.

5. **Scikit-Learn**

This section introduces some packages (or methods) in Python (Scikit-Learn and Scipy) that will be frequently used in your programming assignments. You must become familiar with them and use them masterfully.

- Data Preprocessing (**Documentation Link**)
  - Standardization: StandardScaler
  - Normalization: MinMaxScaler
  - Quantifying Categorical Features: LabelEncoder. OneHotEncoder
  - Construct Train and Test Sets: model_selection.train_test_split
- KNN: KNeighborsClassifier
- Linear Regression: LinearRegression
- Logistic Regression: LogisticRegression, LogisticRegressionCV
- Feature Selection / Model Selection
  - L1 Penalized Regression (Lasso Regression) with Cross-Validation: LassoCV
  - L2 Penalized Regression (Ridge Regression) with Cross-Validation: RidgeCV

  - Cross-Validation: StratifiedKFold, RepeatedKFold, LeaveOneOut, KFold, model_selection.cross_validate, model_selection.cross_val_predict, model_selection.cross_val_score
  - Model Metrics (**Documentation Link**): accuracy_score, auc, f1_score, hamming_loss, precision_score, recall_score, roc_auc_score
- Decision Tree: DecisionTreeClassifier, DecisionTreeRegressor
- Bootstrap, Ensemble Methods
  - Bootstrap: bootstrapped (**Documentation Link**)
  - Bagging: RandomForestClassifier, RandomForestRegressor
  - Boosting: AdaBoostClassifier, AdaBoostRegressor
- Support Vector Machines (**Documentation Link**): LinearSVC, LinearSVR
- Multiclass and Multilabel Classification (**Documentation Link**)
  - One-vs-one Multiclass Strategy: OneVsOneClassifier
  - One-vs-the-rest (OvR) multiclass/multilabel strategy / OneVsRestClassifier
- Unsupervised Learning
  - K-means Clustering: KMeans
  - Hierarchical Clustering: scipy.cluster.hierarchy (not scikit-learn)
- Semi-supervised Learning (**Documentation Link**)

6. **Git and GitHub**

   (a) In the directory of this jupyter notebook file locates, initiate a Git repository.

   (b) Check out a new branch called dev and commit the current notebook within this branch.

   (c) Merge the dev branch to the master branch (the default branch).

   (d) Create a temporary repository (just for practicing and you can delete it later) in GitHub.

   (e) Push new changes in the master branch to the remote repository created in step (d).

   (f) Checkout the dev branch again and do some changes to your notebook, and then repeat step (c) and step (e).

7. **Matplotlib**

   Quick start:https://matplotlib.org/3.1.1/tutorials/introductory/pyplot.html

   (a) Create two one dimensional arrays x and y and plot y vs x, add title, xlabel, ylabel, grid.

   (b) Create multiple arrays and plot them with different styles, add legends, add text/mathematical equations on the plot.

(c) Create multiple subplots, play around with the figure size, text font/size.

(d) Get familiar with get current axis (gca) handle to do the above tasks

(e) Change the limits on x and y axes, use logarithmic axes to plot.

8. **Seaborn**

Quick start:https://seaborn.pydata.org/introduction.html

(a) Use the Salaries.csv file in Pandas section.

(b) Create a dataframe and try to plot it with seaborn.

(c) Perform statistical estimation on the data using seaborn in-built functions - lmplot, catplot, relpolt.

(d) Create axis level functions like boxplot to visualize

(e) Visualize the dataset structure using pairplot and jointplot.