University of Nebraska Lincoln

Practical Machine Learning
ENGR 491/891

Programming Assignment 2

Spring 2022

Linear Regression & K-Nearest Neighbors

Assignment Goals

The goal of this assignment is to solve regression and classification problems using following models.

- **Part A**: Regression Problem – Linear Regression using the Stochastic Gradient Descent algorithm
- **Part B**: Classification Problem – K-Nearest Neighbors

Assignment Instructions

Note: You must use Scikit-Learn to create the models. You are allowed to use Python libraries such as Pandas, NumPy.

i. The code should be written in a Jupyter notebook. Use the following naming convention.
   <lastname1>_ assignment2.ipynb
ii. The Jupyter notebook should be submitted via Canvas.

The programming code will be graded on **both implementation, correctness and quality of the results**.

Score Distribution

ENGR 891: 100 points

Part A

You will perform regression on the following dataset using the Polynomial Regression Model with the Stochastic Gradient Descent (SGD) algorithm (use Scikit-Learn implementation). Your goal is to:

- Minimize the Test dataset's Mean Squared Error
- Maximize the Test dataset's Coefficient of determination R^2 variance score

Note: grading will be based on the quality of the obtained results. Only the model implementation will not earn full credit.

Dataset:
The energy efficiency dataset *EnergyEfficiency.xlsx* is created to perform energy analysis. The dataset comprises 768 samples and 8 features (X1 to X8). It has two real valued target variables (Y1 and Y2), i.e., heating load and cooling load, respectively.

- X1: Relative Compactness
- X2: Surface Area
- X3: Wall Area
- X4: Roof Area
- X5: Overall Height
- X6: Orientation
- X7: Glazing Area
- X8: Glazing Area Distribution
- Y1: Heating Load
- Y2: Cooling Load

For this task you will only predict the heating load. Thus, use Y1 as the label.

Pre-processing: **[5 pts]**

- Load the *.xlsx* file as a Pandas DataFrame object. You may use pandas read_excel() method.
- **Feature Selection**: For optimal test performance, you may use a subset of features. However, it's up to you to decide which features to select or to keep all features.
- Create a separate feature set (Data Matrix X) and target (1D Array y).
- Partition the data in training & test subsets (80% - 20%).

Experiments:

- **Experiment 1**. Create an optimal Polynomial Regression model and train it using SGD with optimal hyperparameters. For the SGD based model selection, see the following notebook: https://github.com/rhasanbd/Linear-Regression-Extensive-

Adventure/blob/master/Linear%20Regression-5-Polynomial%20SGD%20Regressor%20Model%20Selection.ipynb

- Report: degree of the optimal polynomial model and following results for both training and test data: Mean Squared Error & Coefficient of determination R^2 variance score

[15 pts]

- Create a learning curve (negative MSE vs train/validation data) using the optimal model. See how this can be done from block 14 of the following notebook: https://github.com/rhasanbd/Linear-Regression-Extensive-Adventure/blob/master/Linear%20Regression-2-OLS%20Polynomial%20Regression-Frequentist%20Approach.ipynb

[10 pts]

High-Degree (4th Degree) Polynomial Model

In [14]:
```
# Variable that speficies the degree of the polynomial to be added to the feature vector
poly_degree = 4


# Add polynomial and bias term with the feature vector using the sklearn PolynomialFeatures class
poly_features = PolynomialFeatures(degree=poly_degree, include_bias=False)
X_train_poly = poly_features.fit_transform(X_train)
```

Answer the following question.

- Q-1) What does the learning curve tell you? Is your model underfitting or overfitting? Does it have high/low bias and high/low variance? Justify your answer.

[5 pts]

Part B

You will perform binary classification on a COVID-19 misinformation dataset. Your goal is to obtain high precision and recall for the misinformation class. You will use the K-Nearest Neighbors (K-NN) model (use Scikit-Learn implementation).

Note: grading will be based on the quality of the obtained results. Only the model implementation will not earn full credit.

Dataset:
CoAID (Covid-19 heAlthcare mIsinformation Dataset) is a diverse COVID-19 healthcare misinformation dataset, including fake news on websites and social platforms, along with users' social engagement about such news. https://github.com/cuilimeng/CoAID

For this task, you will use the fake and true news articles that are collected from the fact-checked or reliable websites. The true/real and fake news articles are stored in two CSV files: *NewsRealCOVID-19-05.csv* & *NewsFakeCOVID-19-05.csv*

Pre-processing: **[20 pts]**

Following pre-processing steps need to be done:

- Load the two CSV files as two Pandas DataFrame objects.
- Create label columns for each DataFrame object. Use value 1 for true/real news dataset and 0 for fake news dataset.
- Concatenate the two DataFrame objects vertically.
- Extract only two columns for this task.

  Example:

```
df1 = pd.read_csv('/NewsRealCOVID-19-05.csv')
df2 = pd.read_csv('/ NewsFakeCOVID-19-05.csv')
df1['label'] = 1
df2['label'] = 0

df_all = pd.concat([df1, df2], ignore_index=True)
df = df_all.loc[:, ['content', 'label']]
```

- **Exploratory Data Analysis (EDA):** Perform EDA as shown in the following notebook: https://github.com/rhasanbd/Text-Analytics-Beginners-Toolbox/blob/master/Text%20Analytics-III-Classification.ipynb

  Report: include the observations from EDA.

- Perform text normalization by using the lemmatization technique.
- Create a separate feature set (Data Matrix X) and target (1D Array y).
- Partition the data in training & test subsets (80% - 20%).
- Use the same training and test subsets from all experiments.

Before using the K-NN model, you need to vectorize the text features. See how this can be done from the following notebook:
https://github.com/rhasanbd/Text-Analytics-Beginners-Toolbox/blob/master/Text%20Analytics-III-Classification.ipynb

Note that the feature vectorization can be done in multiple ways: using binary features, using frequency of the features, and using the TF-IDF values of the frequency counts. ***You will have to determine the optimal vectorization technique by comparing 3 techniques***.

Finally, the K-NN vectorized data matrices (train & test) need to be standardized before performing model selection.

Report the following results for each experiment:

- Test accuracy
- Test Confusion Matrix
- Test Classification Report

Experiment 2: Perform binary classification using the K-NN. Use hyperparameter tuning to optimize the performance.

[**10 pts**]

Answer the following question.

- Q-2) Why is the performance (precision & recall) of the true/real class higher than the fake class? Explain.

[**5 pts**]

Experiment 3: Generate the ROC curve and the Precision-Recall Curve for the model of experiment 2. Find the optimal threshold. Using the optimal threshold, compute test accuracy, test confusion matrix, and test classification report.

[**10 pts**]

Answer the following question.

- Q-3) Compare the results from experiment 2 and 3. What changes do you observe. Explain.

[**5 pts**]

Experiment 4: Based on the precision-recall curve from experiment 3, find a threshold the increases the test recall for the misinformation class, but does not decrease the test precision much for the same class. Using the optimal threshold, compute test accuracy, test confusion matrix, and test classification report.

[**10 pts**]

Answer the following question.

- Q-4) Compare the results from experiment 3 and 4. What changes do you observe. Explain.

[**5 pts**]

Deliverables:

You will submit two products.

- A single Jupyter notebook containing all experiments. For each experiment show the required performance measures. If it's convenient, you may split your experiments in two notebooks and submit. Your code must be clearly annotated. Also add header descriptions for each block.
- A PDF copy of the written report with the cover page (get it from Canvas). The report must include the following items.

a) Part A: degree of the polynomial of the optimal model and following results for both training and test data: Mean Squared Error & Coefficient of determination R^2 variance score.
b) Part A: learning curve.
c) Part A: Answer to Q-1.
d) Part B: observation from the EDA.
e) Part B: for all experiments (2 to 4) report test accuracy, test confusion matrix and test classification report.
f) Part B: Answer to Q-2, Q-3 & Q-4.