PACE UNIVERSITY SEIDENBERG SCHOOL OF CSIS

CS677- Fall 2025 CRN# 72879
Machine Learning – Final Project
TOTAL POINTS: 100 (Team Project)
DUE DATE: 12/18/2025 (Dec 18th)

Write Python scripts in order to complete the below tasks along with their output. All work should be done and submitted in a single Jupyter (or Colab) Notebook.

1. Select Problem Statement and Dataset: (5 points)

- Identify a specific problem statement for your machine learning project, such as binary classification, multiclass classification, or regression.
- Choose an appropriate dataset that is relevant to your problem statement. Public datasets or datasets that are peer-reviewed are preferred.
- The dataset selected should represent the real-world problem you are trying to solve and have enough data to train and evaluate your models.

2. Perform Exploratory Data Analysis (EDA): (10 points)

- Load and explore the dataset to gain insights.
- Analyze the data distribution and statistical information, identify missing values, outliers, or anomalies, and visualize the relationships (correlation with heat map) between features.
- Perform data preprocessing tasks such as handling missing values, handling categorical variables, and feature engineering (feature scaling, feature selection, feature transformation, feature extraction, feature creation…)

3. Use more than one ML algorithm. Build ML models with different Machine Learning Algorithms to compare them: (10 points)

- Define the inputs (Feature Variable 'X'), and the Label/Target variable 'y'
- Split the dataset into training and testing sets for model training and evaluation (Use 80%-20% ratio) along with the validation set.
- Build at least two models (you can have more than two) using different machine learning algorithms suitable for your problem statement, support vector machines, linear models, etc., and compare their performance with each other.
- Demonstrate any two concepts such as underfitting/overfitting, learning curves, applying Kernel functions, cross-validation (k-fold or stratified) in your project.

4. Use an Optimizer (Gradient Descent or SGD): (5 points)

- Implement gradient descent or SGD algorithm to optimize model parameters and minimize errors between predicted and actual values.

5. Use Hyperparameter Tuning: (At least 2 from the slides discussed) (5 points)

- Tune any two hyperparameters for the algorithms used, such as the number of estimators, learning rate, maximum depth, and other hyperparameters based on your problem statement and dataset to achieve the best

performance.

6. Evaluate Model Performance (Evaluation): (10 points)

- Use appropriate evaluation metrics such as **Accuracy**, **Precision**, **Recall**, **F1-score**, or **Mean Squared Error (MSE)**, **Mean Average Error(MAE),** etc., to evaluate the performance of your models on the test set. Use error vs training set plots to show the model learning trends.

7. Compare Model Performance (Results): (10 points)

- Compare the performance of the different models based on their evaluation metrics, such as accuracy or MSE, to identify the best-performing model.
- Consider factors such as computational efficiency, and model complexity when comparing the models.

8. Project Summary Report:(15 points)

- Prepare a conclusion summary report in word document to explain why you selected a particular ML algorithm for your problem statement. Explain which Algorithm performed well and why it had better results among the other algorithms selected.

9. Presentation: (30 points)

- Record a 12-15 min presentation video by Team and upload to classes (points included). Only one Team member can upload but all Team members must collaborate and present.

10. Libraries:

- Data & Preprocessing: **Pandas**, **Numpy**
- Visualization: **Matplotlib**, Plotly, Seaborn, GGplot, Bokeh…
- ML: **Scikit-Learn**, SciPy
- Neural Nets: Tensorflow, Keras, Theano, PyTorch

11. Links to Datasets:

- https://data.ny.gov/
- https://data.world/city-of-ny?entryTypeLabel=dataset&tab=resources
- https://opendata.cityofnewyork.us/
- https://datasetsearch.research.google.com
- https://archive-beta.ics.uci.edu/
- https://datausa.io/
- https://nces.ed.gov/

Presentation Instructions:

- Record a 12-15 min presentation video and upload to classes
- You can Present your work in PowerPoint presentation and Jupyter Notebook.
- PowerPoint slides must include necessary screenshots of EDA, Evaluation Metrics, and Results/output.
- All Team members must present their contribution.

Submission:

1. Do not zip the files.
2. Submit a video recording of your presentation. Any one Team member can upload.
3. Submit **PPT slides**.
4. Submit a **Jupyter notebook** with a Python script, with comments explaining what each code block means to perform the above task.
5. Submit word document as **Project summary report**

   Do not submit zip files. If you have multiple scripts upload them individually.