CS 559: Machine Learning: Fundamentals and Applications

Assignment 1 Due: 2/3/2025 11:59 p.m.

- The assignment must be individual work and must not be copied or shared. Any tendency to cheat/copy evidence will lead to a 0 mark for the assignment.
- Students must only use Pandas, NumPy, Matplotlib, and Scipy if the problem does not specify libraries/packages.
- All codes will be tested in grading. Any codes with an error will be marked 0. Make sure to restart the kernel and run it all before the submission. Delete any codes you do not want to be graded.
- Results must be displayed.
- All problems must be submitted in a single notebook file. Do not use a text editor to write codes.

1 Linear Regression Modification [40 pts]

In the lecture, the linear regression algorithm, **Linear_Regression(learning rate, iteration)**, was implemented to solve a weight vector in a gradient descent fashion. In this assignment, we will modify the linear regression that optimizes the weights via stochastic gradient descent and predict the house price of the unit area.

a. Load the provided data set as follows.

```
import pandas as pd
df = pd.read_csv('./Realestate.csv')
```

b. [5 pts] Standardize the data using each feature's mean (μ) and standard deviation (σ_i) as shown below

$$\mathbf{x}_{i,std} = \frac{\mathbf{x}_i - \mu_i}{\sigma_i} \tag{1}$$

where i is the feature index. Standardize the target variable as well. The standardized data set will be used for the following questions.

c. [5 pts] Using **Scikit-learn Linear Regression**, predict the target and report the root mean square error value (RMSE).

d. [10 pts] Using the **NumPy** Linear_Regression() implemented in the lecture, predict the target and report the RMSE value. Tune learning rate η and the iteration number so the RMSE value is not more than 10% of the RMSE value from the Scikit-learn linear regression result. Report the final RMSE and η values.

e. [20 pts] Modify the Linear_Regression() algorithm that estimates the weights in a **stochastic gradient descent** manner. Report the RMSE value. Start with the η value found from (d) and tune the parameters until the RMSE value is within 10% of the RMSE value from (c).

2 Bayesian Theorem [15 pts]

a. Load the provided data set as follows.

```
import pandas as pd
df = pd.read_csv('./data.csv')
```

b. [5 pts] Calculate a prior probability of x and y, $p(x = 0)$, $p(x = 1)$, $p(y = 0)$, and $p(y = 1)$.

c. [5 pts] Calculate the likelihood of x, $p(x = 0|y = 0)$, $p(x = 1|y = 0)$, $p(x = 0|y = 1)$, and $p(x = 1|y = 1)$.

d. [5 pts] Calculate the posterior probabilities, $p(y = 1|x = 0)$, $p(y = 0|x = 0)$, $p(y = 1|x = 1)$, and $p(y = 0|x = 1)$.

3 LDA [45 pts]

The binary LDA classification of $\mathbf{X}^{2\times 2}$ was solved by finding the direction of the projection hyperplane ($\mathbf{w} \propto \mathbf{S}_w^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$) in the lecture. In this problem, the binary LDA classification model will be trained and the train model will generalize the test data. Please display the results for full credit consideration.

Generate the data set and split it into train and test sets.

```
from sklearn import datasets
import numpy as np
X, y = datasets.make_blobs(n_samples = 100, n_features = 4, centers = 2,
                           cluster_std= 1.5, random_state= 123)
y = np.array([-1 if i == 0 else 1 for i in y])

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                           test_size=0.2, random_state=42)
```

a. [10 pts] Write a method that computes the between-class scatter matrix $\mathbf{S}_B$. Compute $\mathbf{S}_B$ using a train data.

b. [10 pts] Write a method that computes and returns the within-class scatter of each class, the total within-class scatter ($\mathbf{S}_w$), and its inverse ($\mathbf{S}_w^{-1}$). Calculate $\mathbf{S}_w$ and $\mathbf{S}_w^{-1}$ using the train data.

c. [10 pts] **NumPy.linalg.eig(·)** estimates eigenvalues, λ and their corresponding eigenvectors $\mathbf{u}$. Using **NumPy.linalg.eig(·)**, find the direction of all projection planes, and transform the train data to each projection plane. Determine the most ideal λ and $\mathbf{u}$ for the binary classification. Explain why.

d. [5 pts] Using a choice of $\mathbf{u}$ from 3.c, predict the class of train data and calculate the accuracy score using the scikit-learn accuracy score function.

```
from sklearn.metrics import accuracy_score
```

d. [5 pts] Predict the class of train data using other $\mathbf{u}$ vectors in 3.c and calculate the accuracy. Based on the results in 3.c and 3.d, explain if your choice of $\mathbf{u}$ is the correct decision.

e. [5 pts] Predict the test data class. Report the accuracy. Use scikit-learn LDA to classify the test data.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis(n_components=1)
```

The parameter *n_components* indicates the dimensions that to reduce to.