Homework 1

Due date: Saturday, Oct. 8, 15:59:59 PM, Almaty Time

Programming Assignment: Linear/Logistic Regression, SVM, and Naïve Bayes Implementation

In this programming assignment, you will implement simple linear/logistic regression, SVM and Naïve Bayes classifiers, and run them on a few different datasets. The goal of this assignment is to help you understand the fundamentals of a few classic machine learning methods and become familiar with scientific computing tools in Python. You will also get experience in hyperparameter tuning and using proper train/test data splits.

1. Predict house price of King county in the USA by modifying the linear regression sample code provided in Practice 2 at the class. Dataset and code will be available in the class file directory (15 points).
   A. Linear Regression: Code in Practice 2
   B. Dataset: Dataset in class file directory
   C. The price should be predicted with features of bedrooms, yr_built, and grade.
   D. Performance metric should include training loss.
   E. You should predict the price with the following data (3 bedrooms, Year 1980, grade 8). In implementing linear regression, you may use the codes to load the dataset, which is shown below.

   Programming Tips: Initially, you may limit the row size of loaded data file for fast experimentation using df = df.iloc[:300, :] as shown below. For final evaluation, you may use the entire dataset. In order to make the SGD converge efficiently, you may try to use learning rate decay function: eta = 0.95 * eta. You should extend the code to incorporate the grade feature.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import pandas as pd

# load dataset
df = pd.read_csv("kc_house_data.csv")

# use 300 samples for fast experimentation, for final run, you may
# run the entire dataset
df = df.iloc[:300, :]

# select features and labels for prediction
features = df[["bedrooms", "yr_built"]].values
labels = df[["price"]].values

# normalize data
scaler = MinMaxScaler()
features = scaler.fit_transform(features)
labels = scaler.fit_transform(labels)
```

```
# use the following code for shuffling dataset
concat_data = np.concatenate((features, labels), axis = 1)
np.random.shuffle(concat_data) # shuffle the dataset

X = concat_data[:, 0:3]
y = concat_data[:, 3:4]

# Use the following code to get the predicted price of new house
new_features = np.array([[3, 1980, 8]])
new_features = scaler.transform(new_features)
predicted_value = model.predict(new_features)
print("Predicted price for 3 beds, 1980, grade 8 is:",
int(scaler.inverse_transform(predicted_value)))
```

2. Predicting the survival of Titanic passengers by modifying the logistic regression sample code provided in Practice 2 at the class. You are going to use the famous Titanic dataset. Both the sample code and dataset are available in the class file directory. This is a binary classification problem: Based on passengers' stats, predict whether a passenger will survive from the aground Titan or not. The dataset should be split into training data and test data with 80:20 ratio (15 points).
   A. Logistic Regression: Code in Practice 2
   B. Dataset: Dataset in class file directory
   C. The classification, i.e., survival, is based on sex, age and economic status (Pclass) of the dataset. You should extend the code provided in Practice 2 for classification using these features.
   D. Performance metrics should include the confusion matrix, accuracy score, classification report. You can measure the performance in both training and test data, but submit the performance metrics in test data only.

   Programming Tips: The basic code for implementation of performance metrics is shown below.

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
Y_actual = [1, 1, 0, 1, 0, 0, 1, 0, 0, 0] # this is example data
Y_predic = [1, 0, 1, 1, 1, 0, 1, 1, 0, 0] # this is example data
results = confusion_matrix(Y_actual, Y_predic)
print ('Confusion Matrix :')
print(results)
print ('Classification Report : ')
print (classification_report(Y_actual, Y_predic))
```

   Programming Tips: You may make use of the code shown below for dataset loading and normalization. For label, you may not use normalization since it is 1 or 0.

```
# load dataset
df = pd.read_csv("titanic_data.csv")
```

```
# preprocess dataset by changing the string to integer, and filling
in the missing values
df['Sex'] = df['Sex'].map({'female':1,'male':0})
df['Age'].fillna(value=df['Age'].mean(), inplace=True)

# initially experiment with 100 samples. For final run, use full
# dataset
#df = df.iloc[:100, :]
# select proper features for prediction
passengers = df[["Sex", "Age", "Pclass","Survived" ]]

# split train and test set
train, test = train_test_split(passengers, test_size=0.2)
# select proper features for prediction
train_features = train[["Sex", "Age", "Pclass"]].values
test_features = test[["Sex", "Age", "Pclass"]].values
train_labels = train[["Survived"]].values
test_labels = test[["Survived"]].values

# normalize data
#scaler = StandardScaler()
scaler = MinMaxScaler()
train_features = scaler.fit_transform(train_features)
test_features = scaler.fit_transform(test_features)

# shuffle features and labels of training data
concat_data=np.concatenate((train_features, train_labels), axis = 1)
np.random.shuffle(concat_data) # shuffle the training dataset

# divide shuffled dataset to features X and labels y
X = concat_data[:, 0:3]
y = concat_data[:, 3:4]
```

Programming Tips: For measuring the performance, `y_predic` is obtained below.

```
# Performance measure

# use the test features
X = test_features
y = test_labels
w = model.weights # use the weights resulting from training
y_predic = []
for j in range(len(X)):
    model = LogisticRegression(X[j], w, y[j])
    if model.predict(X[j]) >= 0.5:
        y_predic.append(1)
    elif model.predict(X[j]) < 0.5:
        y_predic.append(0)
```

3. In this problem, you will implement multiclass-classification using Scikit library. The dataset is Iris dataset. Here, the dataset should be split into training data and test data with 70:30 ratio. You will implement multi-class classification using logistic regression, Naïve Bayes classifier, and Gaussian RBF of SVM. You will compare the performances of these classifiers in terms of accuracy and confusion matrix, recall, precision and F1-score (30 points).

   A. Dataset: Iris dataset of Practice 1
   B. Logistic Regression (LogistRegression_P3.py)
   C. SVM with one-vs-all approach (SVM_P3.py)
   D. Naïve Bayes (NB_P3.py)
   E. Performance metrics: accuracy, confusion matrix, recall, precision and F1-score
   F. Comparative performance evaluation of logistic regression, SVM and NB for multi-class classification

In implementing logistic regression, you may use one-vs-rest for multi-class classification. In implementing SVM, you should use Gaussian RBF with one-vs-rest approach for breaking down the multiclass problem into multiple binary classification problems. The sample codes using scikit-learn are provided in Homework 1 directory in MS Teams. You should comparatively analyze the performances of the above classifiers in terms of accuracy and confusion matrix, recall, precision and F1-score. Please change the hyperparameters which you have used for training each model, and try to get the best performance for each model, and submit them. You should explain how the best performances are achieved. Please describe your analysis in the report template provided in Homework 1 directory. All features of Iris dataset should be used in training and for performance evaluation in testing.

Programming Tips: The sample codes will be provided and will be available in the class file directly of MS teams. The sample codes will be given after the relevant topics are taught in the class. For RBF SVM implementation, you may refer to the scikit site(https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html) for adjusting the hyperparameters.

Extra Credit (10 points)

Predict the class of wine using the Wine dataset available at UCI open dataset (https://archive.ics.uci.edu/ml/datasets/wine), with dataset split of 70:30. You should use all features of Wine dataset in getting the accuracy. You should try all 3 models as described above, i.e., Logistic Regression, SVM, and Gaussian RBF for your experimentation, but submit only one model which gives you the best accuracy score.

In the report, you should explain how the best accuracy is obtained, i.e., the hyperparameters' setting and any other techniques that you have applied to get the high accuracy. The highest extra credit will be given to the student who got the highest accuracy, and the extra credit will be adjusted in accordance with the ranking in accuracy performance.

Programming Environment

If you will be completing the assignment on a local machine then you will need a Python environment set up with the appropriate packages. We suggest that you use Anaconda to manage Python package dependencies (https://www.anaconda.com/download). This guide provides useful information on how to use Conda: https://conda.io/docs/user-guide/getting-started.html. You also need to set up sci-kit learn tool. For information on how to install and use sci-kit learn, visit https://scikit-learn.org/stable/.

None of the parts of this assignment require the use of a machine with a GPU. If you would like to use GPU, you may use Google Colaboratory. Whether you are using local machine or Google Colab, ensure that IPython is installed (https://ipython.org/install.html). You may code and develop your program using either Google Colab, MS VS code or Eclipse IDE environment. You have to use the Jupyter notebook for submitting your program.

Submission Instructions

You must submit all files to MS Teams Assignment 1. Any inquiry about programming assignment can be sent to Yerkhat Shaukhatov(y_shaukhatov@kbtu.kz), Gulmira Daribayeva (G_Daribayeva@kbtu.kz), and Barganaev Zholdibay Karimuly(zh_barganaev@kbtu.kz) with cc to jtpark2010@gmail.com. In your ipynb file, you may describe the requirement, your code modification, hyperparameter settings, software design architecture, training, test and prediction. Submission of this programming assignment will involve two files:

① All of your code (Python files and ipynb file with description) **in a single ZIP file**. The filename should be **studentid_hw1_code.zip.**

② A brief report **in PDF format** using brief report template provided in the class. The filename should be **studentid_hw1_report.pdf.**

Late submission is not permitted except for the inevitable case in which the grade will be discounted accordingly. The code which does not run successfully will receive no points in grading. Please refer to Academic Honesty in Syllabus on collaborations, late submission, and extension requests.

Problem Set 1(Each problem is 1 point)

Submit answers in a file: **studentid_hw1_answers.pdf.**

You could write your answer by hand on white paper, take it with smartphone camera, and submit it. In this case, please make sure that your answer should be clearly shown.

① Let $f(x) = 5x^2 + 3$. What is the derivative of $f(x)$?

② Let $f(x) = 3e^{2x}$. What is the derivative of $f(x)$?

③ Prove that the derivative of $\sigma(x)$ is $\sigma(x)\big(1 - \sigma(x)\big)$ where $\sigma(x) = \frac{1}{1+e^{-x}}$.

④ Let $f(x, y) = 3xy^2 + 2y$. What is the following partial derivative $\frac{\partial}{\partial x} f(x, y)$?

⑤ Let $f(x, y) = 3xy^2 + 2y$. What is the following partial derivative $\frac{\partial}{\partial y} f(x, y)$?

The chain rule in differentiation is as follows: If $y = \mathrm{f}(u)$ and $u = g(x)$ are both differentiable functions, then $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u}\frac{\partial u}{\partial x}$

⑥ Let $y = (o - t)^2$ and o $= w_1x_1 + w_2x_2 + w_3x_3$. $\frac{\partial y}{\partial x_3}$ = ?

⑦ Let $y = -c\ln o + (1 - c)\ln(1 - o)$ and o= σ(x). $\frac{\partial y}{\partial x}$ = ?

Basic matrix multiplication formula is described below

A. Matrix-Vector Product: $\begin{bmatrix} a & b \end{bmatrix} \bullet \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg \; af + bh \end{bmatrix}$

B. Inner Product: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$

C. Outer Product: $\begin{bmatrix} a \\ b \end{bmatrix} \bullet \begin{bmatrix} c & d \end{bmatrix} = \begin{bmatrix} ac & ad \\ bc & bd \end{bmatrix}$

D. Transpose: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

⑧ What is $\begin{bmatrix} 1 & 3 \end{bmatrix} \bullet \begin{bmatrix} 3 & 5 \\ 6 & 2 \end{bmatrix}$?

⑨ What is $\begin{bmatrix} 2 & 1 \\ 5 & 2 \end{bmatrix} \bullet \begin{bmatrix} 3 & 5 \\ 6 & 2 \end{bmatrix}$?

⑩ Let matrix A = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$. What is transpose of A ? A^{T} = ?

⑪ A fair coin has equal probability of coming up heads or tails. If a fair coin is flipped twice, what is the probability of seeing first heads, then tails: Pr(heads, tails) = ?

⑫ Let you flip the fair coin one time and you receive $1 if it comes up heads and $2 if it comes up tails. Let X be the number of dollars you receive which is a *random variable.* What is the expected value of X ? $E[X]$ = ?

⑬ Let vectors u and v are defines as follows: $\mathrm{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mathrm{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$. What is inner product of vectors u and v ?

⑭ What is L2 norm of u ?

⑮ How much time did you spend to solve the above mathematical questions ?

Problem Set 2: For the predicted value of binary classifier, y_prediction, and actual value y_actual, calculate performance metrics shown below (9 points)

y_actual	y_prediction	output with threshold 0.6
0	0.5	0
1	0.9	1
0	0.8	1
0	0.2	0
1	0.3	0
0	0.6	1
1	0.4	0

Calculate confusion matrix:

TP:	FP:
FN:	TN:

Accuracy:

Recall:

Precision:

F1-Score:

Specificity: