CS1: Selected Topics in Computer Science

(COVER SHEET)

Project Name: Diabetes prediction using logistic regression

Team ID: 51

Team Information:

	ID [Ordered by ID]	Full Name [In Arabic]	Attendance [Handwritten Signature]	Final Grade
1	201901009	أحمد عيد فوزي سيد		
2	202000223	تسنيم سامح سليمان محمود		
4	202000945	منة الله محمد سيد عيسوى		
6	202001066	يمنى محمد عبد القادر عبد الغني		

Table of Contents:

Project Description Document:

Dataset Description

Image Dataset:

Name: Plant Pathology

https://www.kaggle.com/c/plant-pathology-2020-fgvc7/data

Brief: Identify the category of foliar diseases in apple trees

Description: Given a photo of an apple leaf, can you accurately assess its health? This competition will challenge you to distinguish between leaves which are healthy, those which are infected with apple rust, those that have apple scab, and those with more than one disease.

Train.csv
image_id: the foreign key
combinations: one of the target labels
healthy: one of the target labels
rust: one of the target labels
scab: one of the target labels
images
A folder containing the train and test images, in jpg format.

test.csv
image_id: the foreign key

sample_submission.csv
image_id: the foreign key
combinations: one of the target labels
healthy: one of the target labels
rust: one of the target labels
scab: one of the target labels

Total number of samples: 3640
Samples for Training: 1820
Samples for Testing: 1820

Numerical Dataset:

Name: Diabetics prediction

https://www.kaggle.com/datasets/kandij/diabetes-dataset?select=diabetes2.csv

Brief: Predicting whether the person is having diabetes or not.

Description: The data was collected and made available by "National Institute of Diabetes and Digestive and Kidney Diseases" as part of the Pima Indians Diabetes Database. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here belong to the Pima Indian heritage (subgroup of Native Americans), and are females of ages 21 and above.

diabetes2.csv
Pregnancies: Integer
Glucose: Integer
BloodPressure: Integer
SkinThickness: Integer
Insulin: Integer
BMI: Decimal
DiabetesPedigreeFunction: Decimal
Age: Integer
Outcome: Target Label

Total number of samples: 768

Implementation & Results

SVM MODEL (Image Dataset)

Implementation

Size of Image: 100

Samples for Training: 1456

Samples for Validation: 365

Samples for Testing: 1820

Features:

Number of extracted Features : 30000 features

The dimension of resulted features : [1456 , 30000]

Cross Validation:
Training/Validation Ratio : 0.20

Hyperparameters used:
Kernel = 'linear'
Gamma = 0.001
Probability = True

Results:

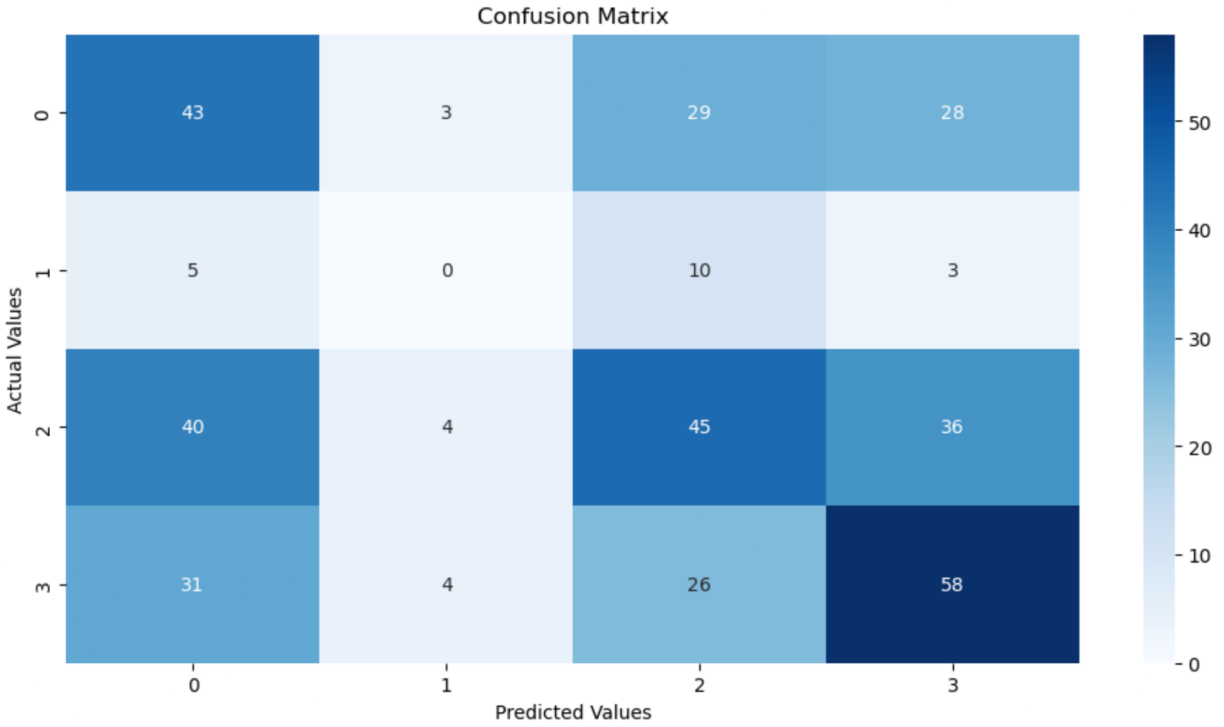
Accuracy:

```
from sklearn.metrics import accuracy_score
print(f"The model is {accuracy_score(y_pred,y_test)*100}% accurate")
```

✓ 0.5s Python

```
The model is 40.0% accurate
```

Confusion Matrix:



AUC:

```
from sklearn.metrics import roc_curve, auc
print("\nArea Under Curve (auc): {}".format(round(metrics.roc_auc_score(y_test, y_pred_prob, multi_class="ovr",avera
print("\n")
print(metrics.classification_report(y_test , y_pred))
```

✓ 0.5s Python

```
Area Under Curve (auc): 0.6136


              precision    recall  f1-score   support

           0       0.36      0.42      0.39       103
           1       0.00      0.00      0.00        18
           2       0.41      0.36      0.38       125
           3       0.46      0.49      0.48       119

    accuracy                           0.40       365
   macro avg       0.31      0.32      0.31       365
weighted avg       0.39      0.40      0.40       365
```

ANN MODEL (Image Dataset)

Implementation

Size of Image: 224

Samples for Training: 1092

Samples for Validation: 729

Samples for Testing: 1820

Features:

Number of extracted Features : 150528 features

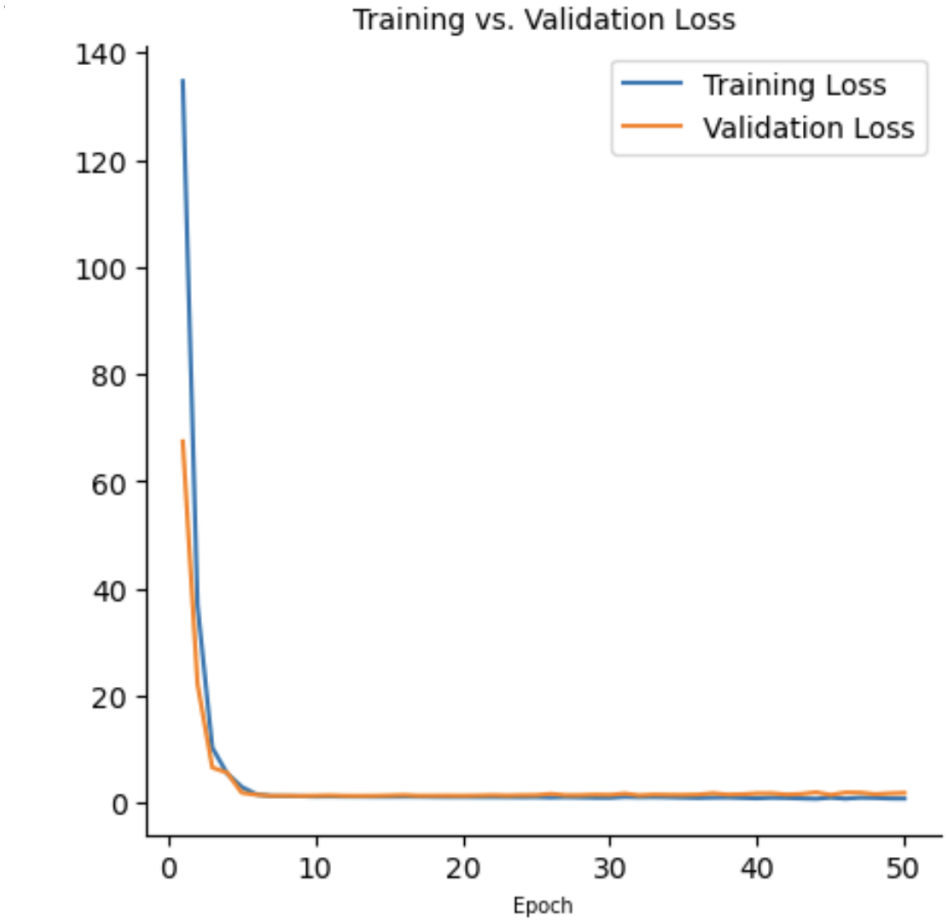
The dimension of resulted features : [1092, 15052]

Cross Validation:
Training/Validation Ratio : 0.40

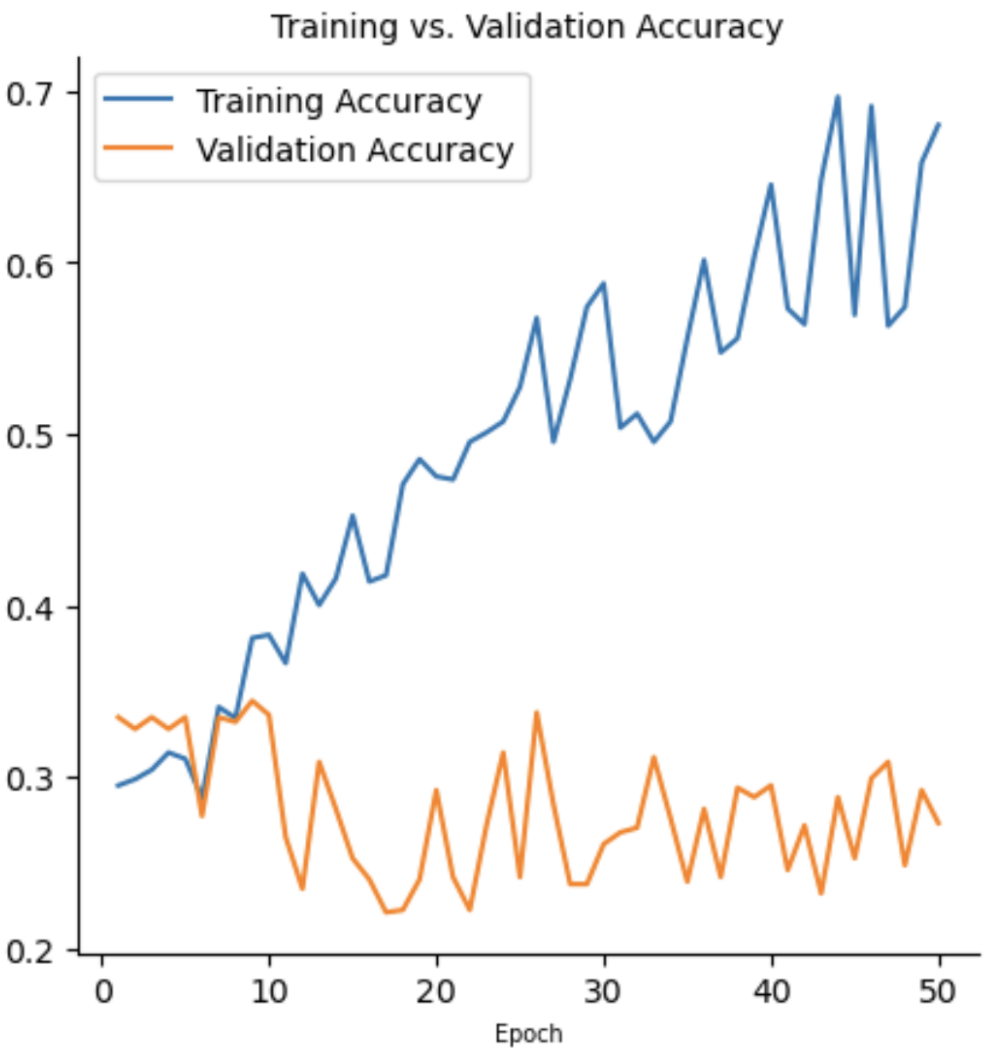
Hyperparameters used:
Losses = Catogorical_crossentropy
Optimizer = Adam
Learning Rate = 0.001
Catogorical_accuracy = 'accuracy'
Epochs = 50
Batch_Size = 128

Results:

Loss curve



Accuracy Curve




```
35/35 [==============================] - 16s 437ms/step - loss: 0.7435 - accuracy: 0.7198
[0.743474006652832, 0.7197802066802979]
23/23 [==============================] - 10s 431ms/step - loss: 1.8761 - accuracy: 0.2730
[1.8760501146316528, 0.27297666668891907]
```

Logistic Regression Model (Numerical Dataset)

Implementation

Samples for Training: 576
Samples for Testing: 192

Results:

Loss Function:

```
# Running Log loss on training
print("The Log Loss on Training is: ", log_loss(Y_train, pred_proba))

# Running Log loss on testing
pred_proba_t = model.predict_proba(X_test)
print("The Log Loss on Testing Dataset is: ", log_loss(Y_test, y_pred_proba))
```

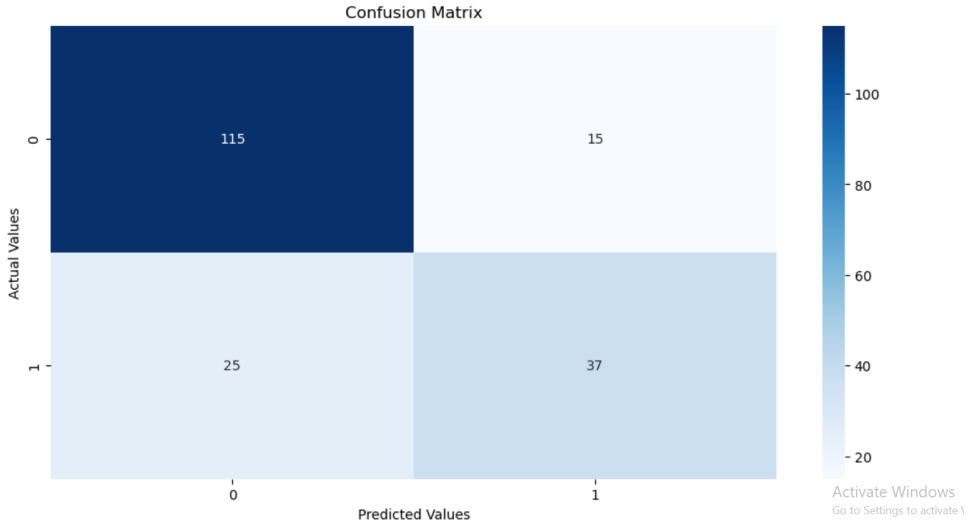
54] ✓ 0.1s

```
The Log Loss on Training is:  0.4843136805007716
The Log Loss on Testing Dataset is:  0.44402298946278146
```

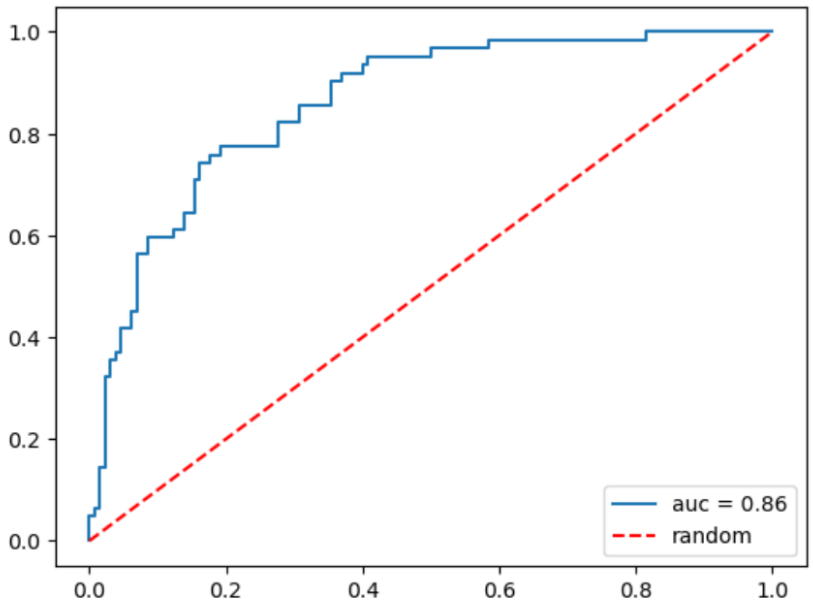
Accuracy:

```
Accuracy for test set is 0.7917.
Precision for test set is 0.7115.
Recall for test set is 0.5968.
              precision    recall  f1-score   support

           0       0.82      0.88      0.85       130
           1       0.71      0.60      0.65        62

    accuracy                           0.79       192
   macro avg       0.77      0.74      0.75       192
weighted avg       0.79      0.79      0.79       192
```

Confusion Matrix



ROC Curve



SVM Model (Numerical Dataset)

Implementation

Samples for Training: 768
Samples for Testing: 154

Cross Validation
Training/Validation Ratio : 0.20

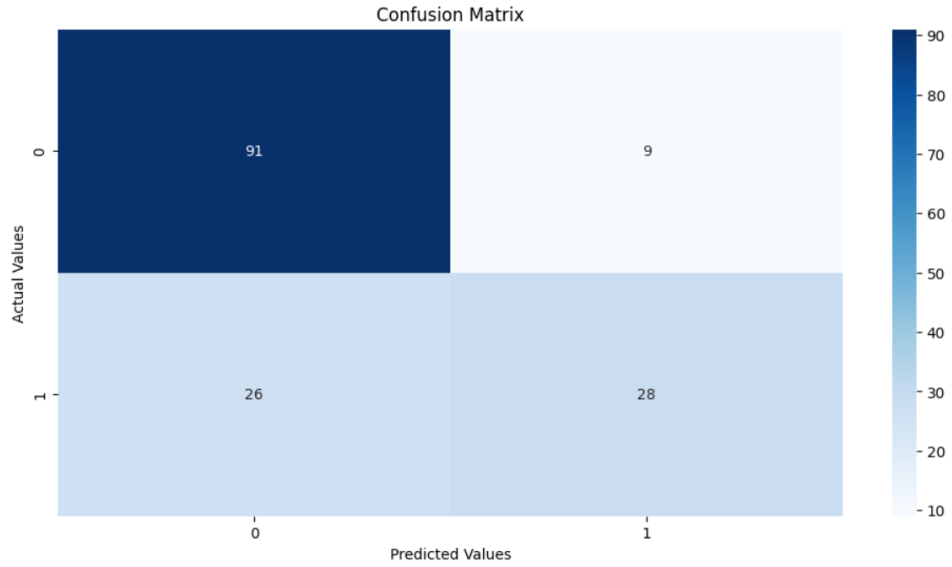
Hyperparameters used:
Kernel = 'linear'

Results:

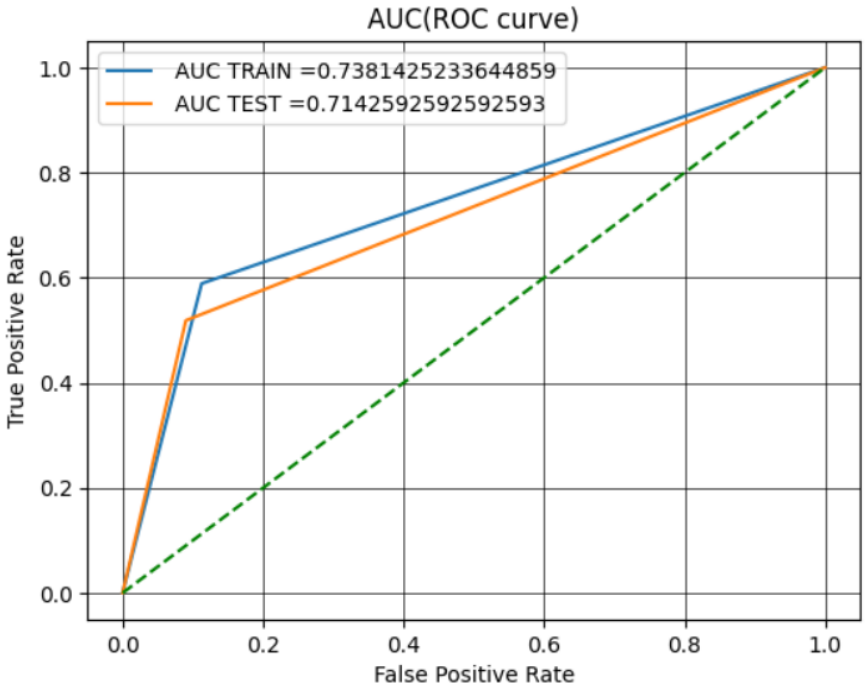
Accuracy:

```
Testing Metrics
Accuracy for test set is 0.7727.
Precision for test set is 0.7568.
Recall for test set is 0.5185.
F1:  0.6153846153846154
Area Under Curve (auc):  0.7142592592592593
              precision    recall  f1-score   support

           0       0.78      0.91      0.84       100
           1       0.76      0.52      0.62        54

    accuracy                           0.77       154
   macro avg       0.77      0.71      0.73       154
weighted avg       0.77      0.77      0.76       154
```

Confusion Matrix


Confusion Matrix
91
9
26
28
0
1
Actual Values
Predicted Values
90
80
70
60
50
40
30
20
10

ROC Curve


AUC(ROC curve)
AUC TRAIN =0.7381425233644859
AUC TEST =0.7142592592592593
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0