SELECTED PROJECT

Supervised By: Dr. Wessam EL Behaidy

Faculty of Computer Science and Artificial intelligence – Helwan Uni.

CS395: Selected cs1

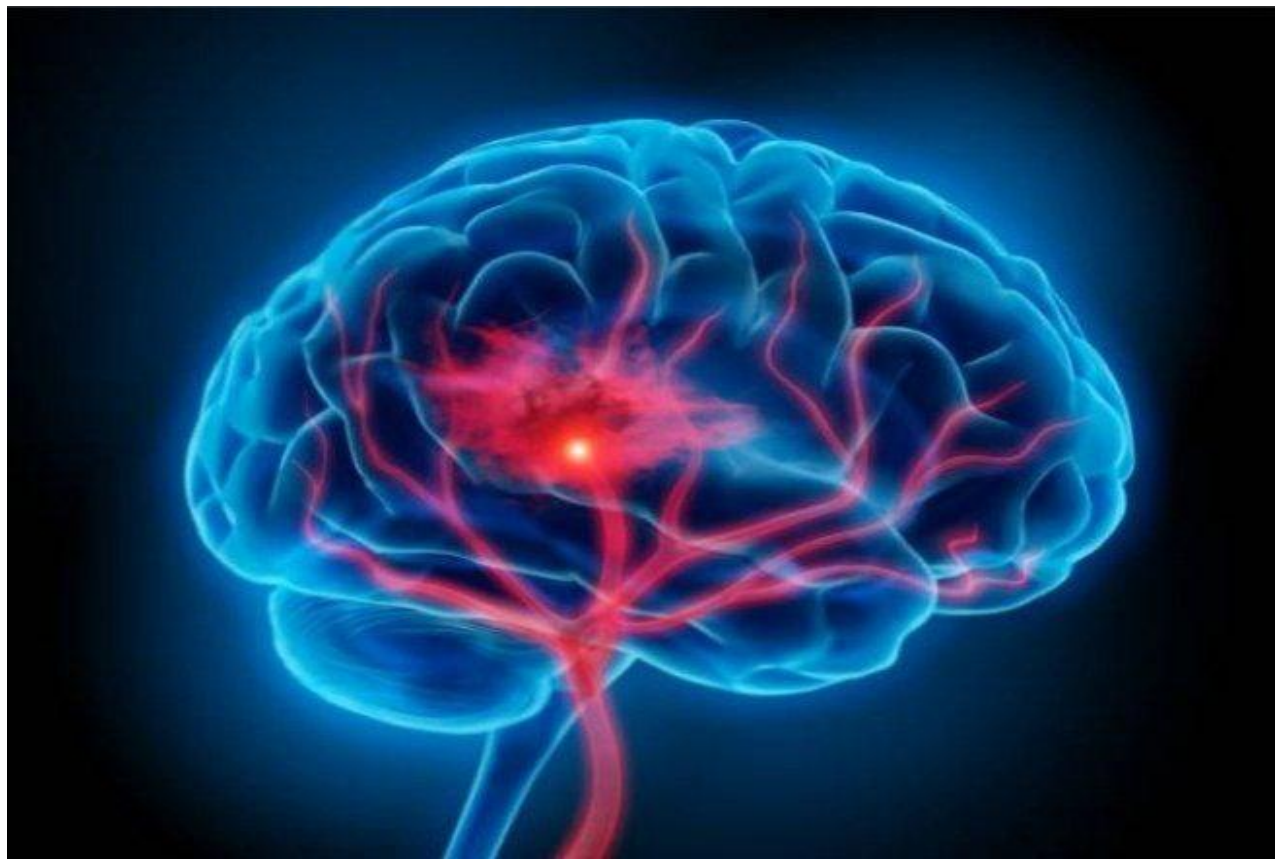
SELECTED PROJECT

Part One

Numerical Dataset

Name Dataset

Stroke Prediction



Link Dataset

https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset [1]

Contents

1) General Information on dataset

Informaion on dataset :

The dataset for stroke prediction is from Kaggle [1].
This particular dataset has 5110 rows and 12 columns.
It is a **binary classification** problem with multiple numberical and categorical features.

Problem Statement :

According to the World Health Organization (WHO), stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. It is another health issue that has found to be rising throughout the world due to the adoption of lifestyle changes that disregards healthy lifestyle & good eating habits. Thus, new emerging electronic devices that record the health vitals have paved the way for creating an automated solution with AI techniques at it's core. Thus, similar to heart diseases, efforts have begun to create lab tests that predict stroke. The dataset presented here has many factors that highlight the lifestyle of the patients and hence gives us an opportunity to create an AI-based solution for it.

Aim :

- To classify / predict whether a patient can suffer a stroke.
- It is a **binary classification** problem with multiple numerical and categorical features.

2) Implementation Details

==>At feature extraction phase, how many features were extracted, their names, the dimension of resulted features

Dataset Attributes :

- **id**: unique identifier
- **gender**: "Male", "Female" or "Other"
- **age**: age of the patient
- **hypertension**: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
- **heart_disease**: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
- **ever_married**: "No" or "Yes"
- **work_type**: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
- **Residence_type**: "Rural" or "Urban"
- **avg_glucose_level**: average glucose level in blood
- **bmi**: body mass index
- **smoking_status**: "formerly smoked", "never smoked", "smokes" or "Unknown"*
- **stroke**: 1 if the patient had a stroke or 0 if not

Each feature will be discussed separately

Figure 1: The association between the gender and age features and the stroke feature expresses the percentage of women and men who had a stroke is approximately 23% and 26%, respectively. That shows that men are by 3% more prone to stroke disease, which, however, still targets men and women.
illustrates the participants' distribution in each class in terms of the age group that they belong to and the gender of each participant. Focusing on the stroke class, in the left fifigure, a signifificant percentage of the participants are older than 70 years, whereas the second, most frequently occurring age group is 50–80. In addition, in this fifigure, we see that stroke mainly concerns elderly people.

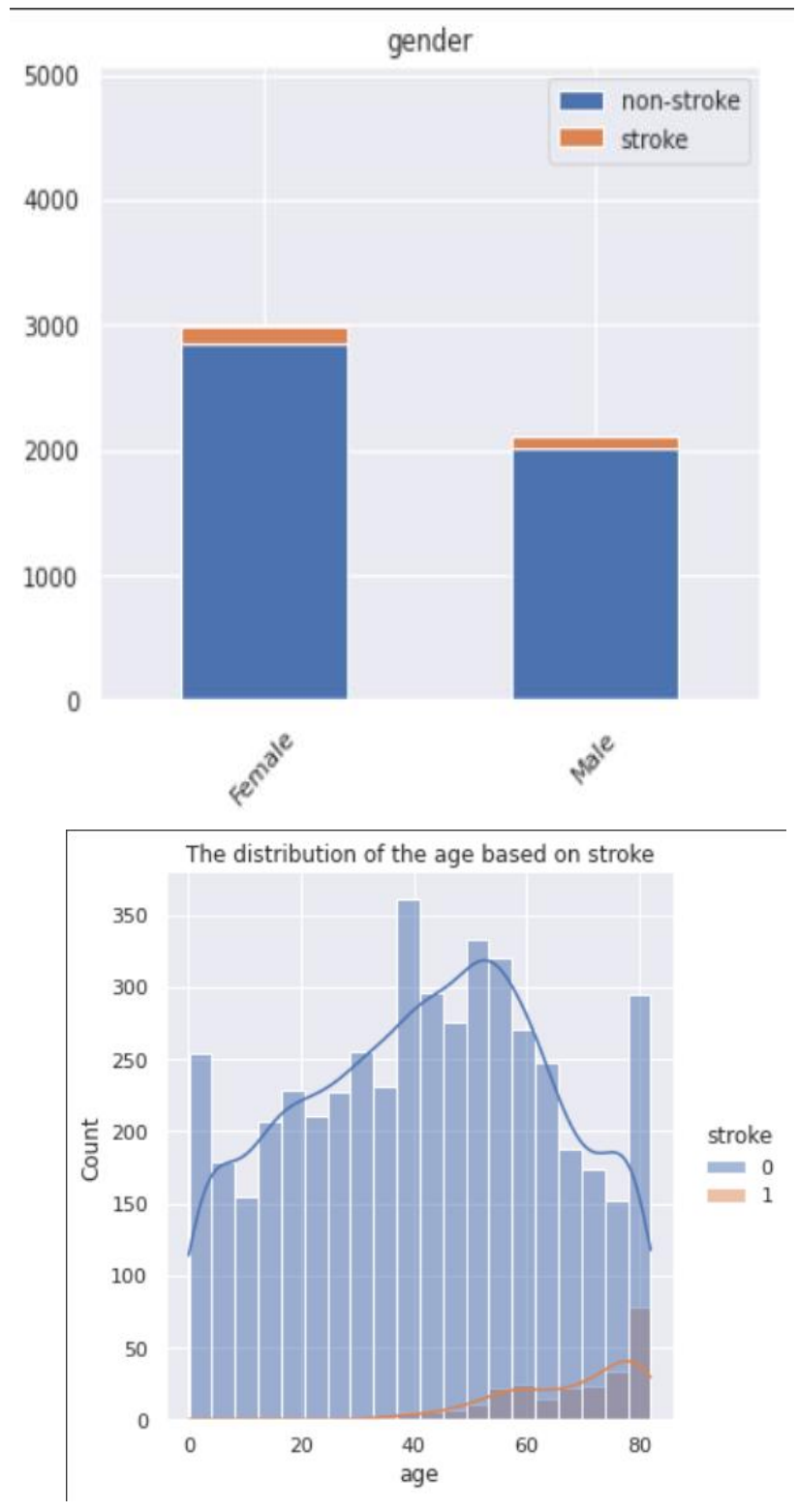


Figure 1. Participants distribution per age group and gender type in the dataset

==> Is cross-validation used in any of the implemented models? If yes, specify the number of fold and ratio of training/validation.

No used cross-validation in any of the implemented model

==> Hyperparameters used in your model, as initial learning rate, optimizer, regularization, batch size, no. of epochs, ect....

--> In logistic Regression :

- Hyperparameters

1- Penalty : default='l2' ,add a L2 penalty term and it is the default choice.

2- Dual : default=False

3- C : float, default=1.0 , Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.

4- fit_intercept : bool, default=True Specifies if a constant (a.k.a. bias or intercept) should be added to the decision function.

- Batch size(the number of training examples utilized in one iteration)

3926 of data size

--> Support Vector Machine (SVM) :

- Hyperparameters (The main hyperparameter of the SVM is **the kernel**. It maps the observations into some feature space.)

In The model we used three different Kernel

(linear)

C=10

Gamma = 1000

Probability = True

Batch size(the number of training examples utilized in one iteration)

3926 of data size

---

3) Results Details

==> For each model you should show all these results for your model on testing data (loss curve, accuracy, confusion matrix, ROC curve)

--> In logistic Regression :

- Accuracy

```
# accuracy on test data
X_test_prediction = log_reg.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, y_test)
print('Accuracy on Test data : ', test_data_accuracy * 100)
```

```
Accuracy on Test data :  79.90605427974948
```

- Precision && Recall Model

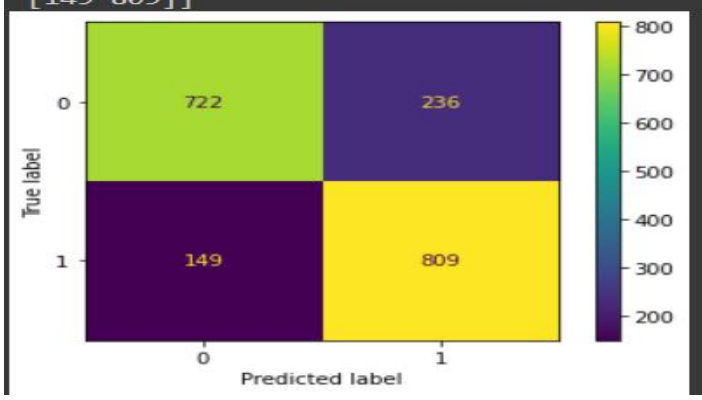
```
# Model Precision: what percentage of positive tuples are labeled as such?
print("Precision:" ,metrics.precision_score(y_test, X_test_prediction))

# Model Recall: what percentage of positive tuples are labelled as such?
print("Recall:" ,metrics.recall_score(y_test, X_test_prediction))
```

```
Precision: 0.7741626794258373
Recall: 0.8444676409185804
```

- Confusion Matrix && Confusion Matrix Curve

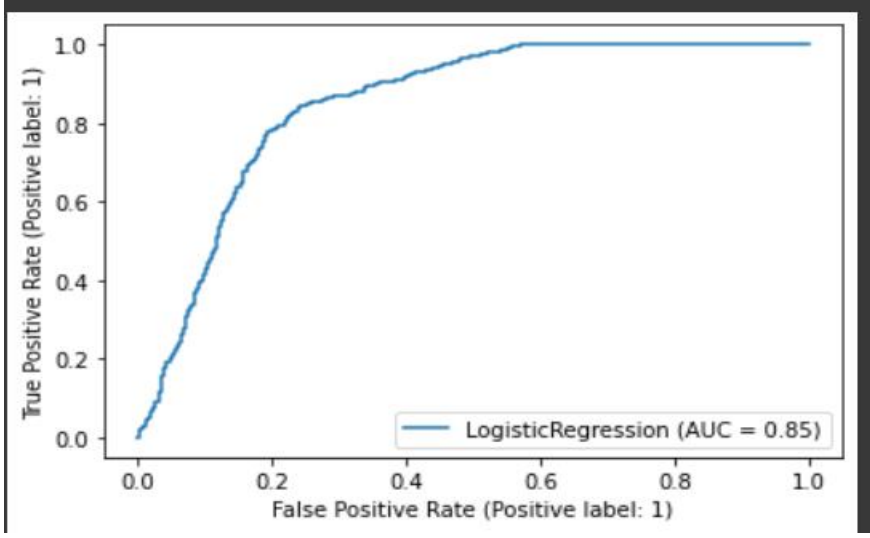
```
#Seaborn Confusion Matrix
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

cf_matrix = confusion_matrix(y_test, X_test_prediction)

print(cf_matrix)
plot_confusion_matrix(log_reg, X_test, y_test)
plt.show()
```

```
[[722 236]
 [149 809]]
```



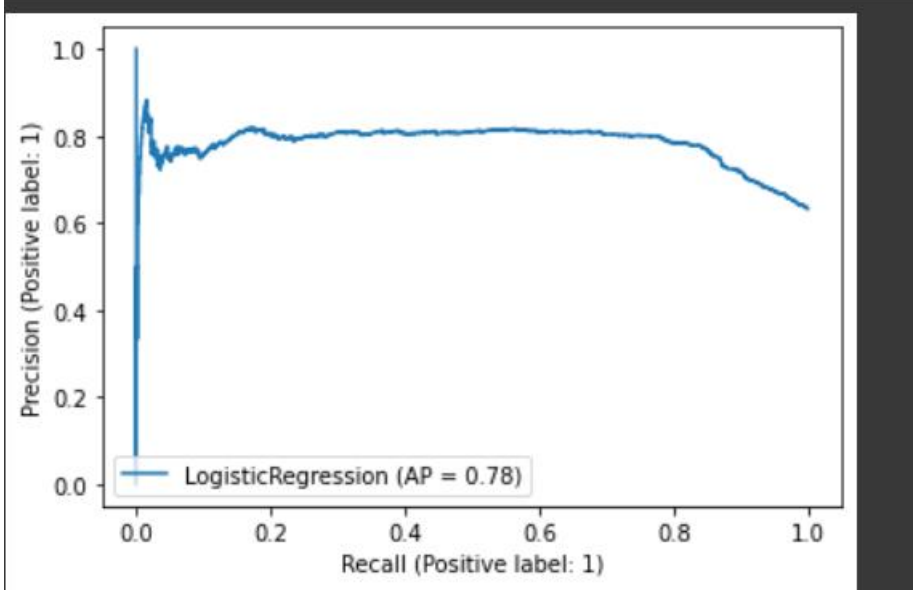
- ROC Curve

```
# ROC (Receiver Operating Curve) and AUC (Area Under Curve)
plot_roc_curve(log_reg, X_test, y_test);
```



- PR Curve

```
#PR curve
plot_precision_recall_curve(log_reg, X_test, y_test);
```



--> Support Vector Machine (SVM)

- Accuracy

```
# Make the predictions
y_predict = svc.predict(X_test_std)

# Measure the performance
from sklearn import metrics
print("Accuracy score %.3f" %metrics.accuracy_score(y_test, y_predict))
```

```
Accuracy score 0.790
```

- Precision && Recall Model

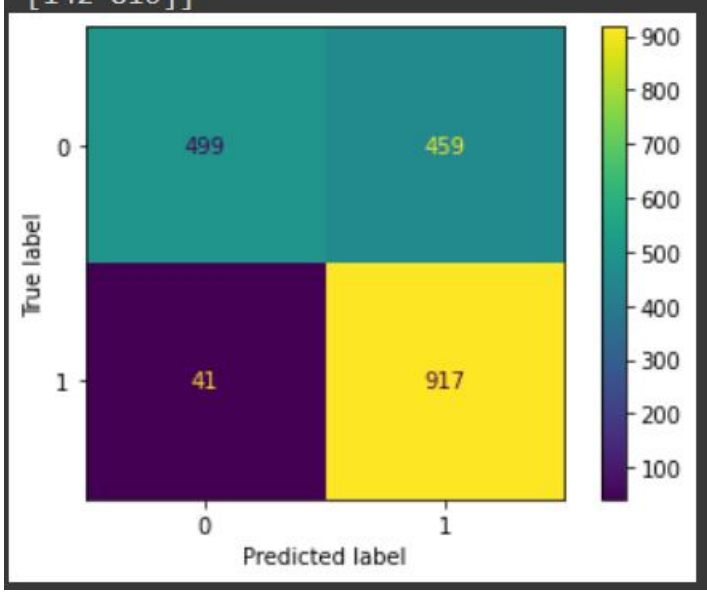
```
# Model Precision: what percentage of positive tuples are labeled as such?
print("Precision:" ,metrics.precision_score(y_test, y_predict))

# Model Recall: what percentage of positive tuples are labelled as such?
print("Recall:" ,metrics.recall_score(y_test, y_predict))
```

```
Precision: 0.7576601671309192
Recall: 0.8517745302713987
```

- Confusion Matrix && Confusion Matrix Curve

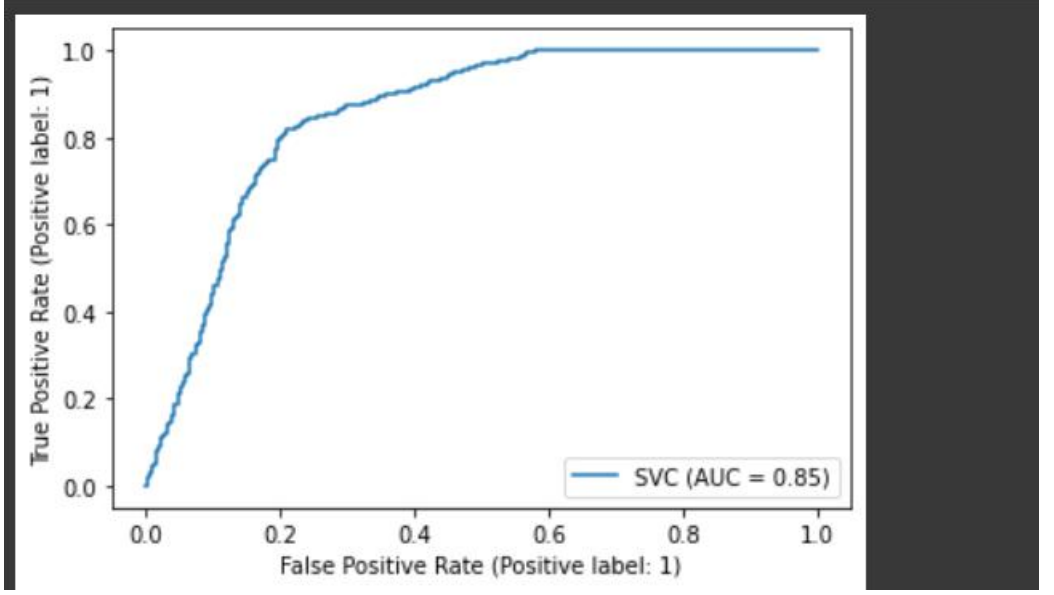
```
#Seaborn Confusion Matrix
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import train_test_split

cf_matrix = confusion_matrix(y_test,y_predict)
print(cf_matrix)
plot_confusion_matrix(svc, X_test, y_test)
plt.show()
```

```
[[697 261]
 [142 816]]
```



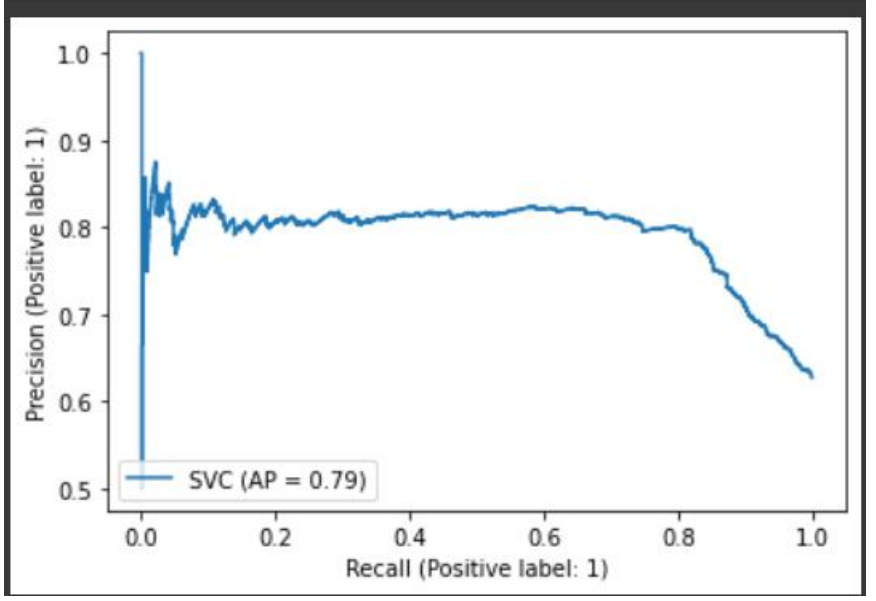
- ROC Curve

```
#ROC (Receiver Operating Curve) and AUC (Area Under Curve)
plot_roc_curve(svc, X_test, y_test);
```



- PR Curve

```
plot_precision_recall_curve(svc, X_test, y_test);
```



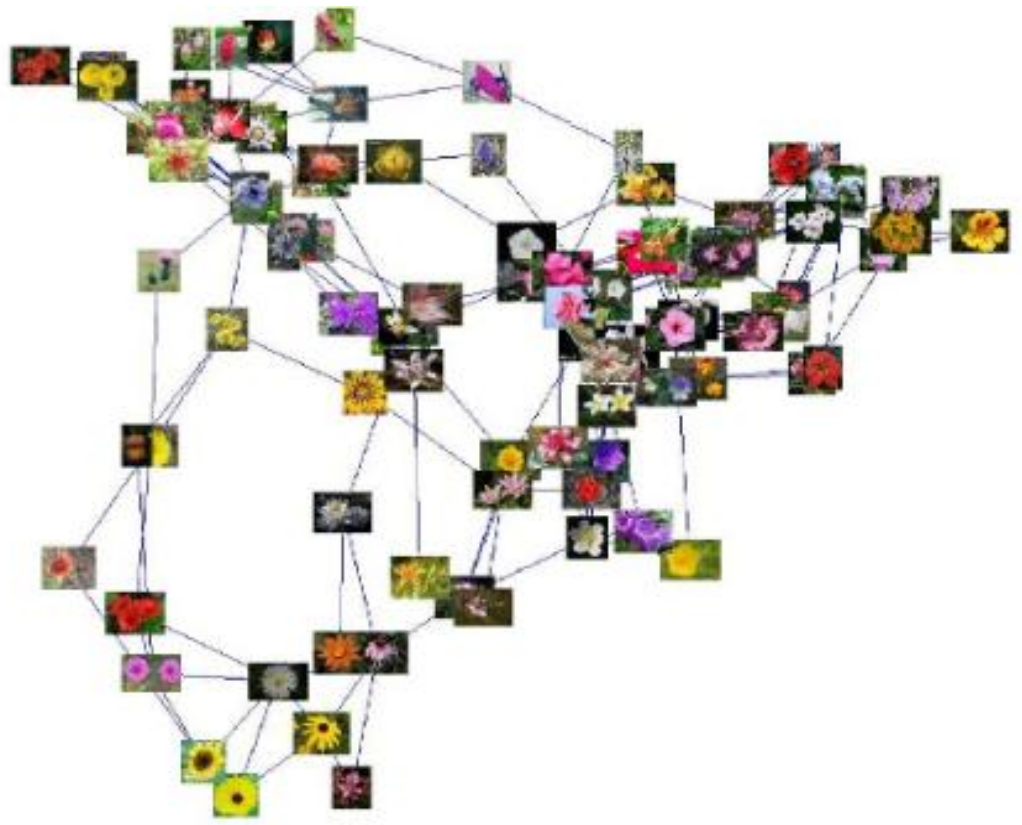
SELECTED PROJECT

Part Two

Image Dataset

Name Dataset

Oxford 102 Flower Dataset



Link Dataset

https://www.kaggle.com/datasets/nunenuh/pytorch-challange-flower-dataset [2]

Contents

1) General Information on dataset

Informaion on dataset :

The Oxford 102 Flower dataset for stroke prediction is from Kaggle **[2].**

It is a **multiple classification** problem.

Problem Overview :

Oxford 102 Flower is an image classification dataset consisting of 102 flower categories. The flowers chosen to be flower commonly occurring in the United Kingdom. Each class consists of between 40 and 258 images.

The images have large scale, pose and light variations. In addition, there are categories that have large variations within the category and several very similar categories.

Numder of samples using in all data = 1068

Numder of samples using in training data = 961

Numder of samples using in testing data = 107

Aim :

The goal of the dataset is to predict which type of rose it will belong to

2) Implementation Details

==>At feature extraction phase, how many features were extracted, their names, the dimension of resulted features

- Different features are chosen to describe different properties of the flower. The low level features we use are colour, histogram of gradient orientations (HOG), and SIFT sampled on both the foreground region and its boundary

==> Hyperparameters used in your model, as initial learning rate, optimizer, regularization, batch size, no. of epochs, ect....

--> Support Vector Machine (SVM) :

- Hyperparameters (The main hyperparameter of the SVM is **the kernel**. It maps the observations into some feature space.)

In The model we used three different Kernel

(linear)

C = 0.1

Gamma= 'scale'

learning_ratestr = 'optimal'

Batch size(the number of training examples utilized in one iteration)

107 of data size

--> Artificial neural network (ANN) :

Optimizer = 'adame'

Epoches = 100

Batch Size = 107

---

2) Results Details

==> For each model you should show all these results for your model on testing data (loss curve, accuracy, confusion matrix, ROC curve)

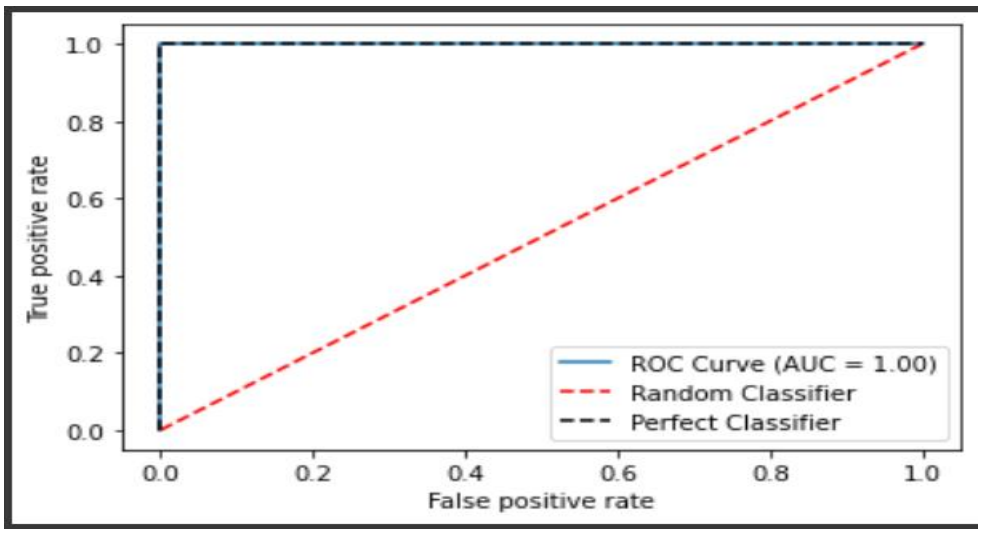
--> Support Vector Machine (SVM)

- Accuracy

```
# Accuracy Score of Testing Data
accuracy = model.score(dx_test_dataset,y_test)
print("Testing  Accuracy -> ", accuracy*100)
```

```
Testing  Accuracy ->  64.48598130841121
```

- ROC Curve



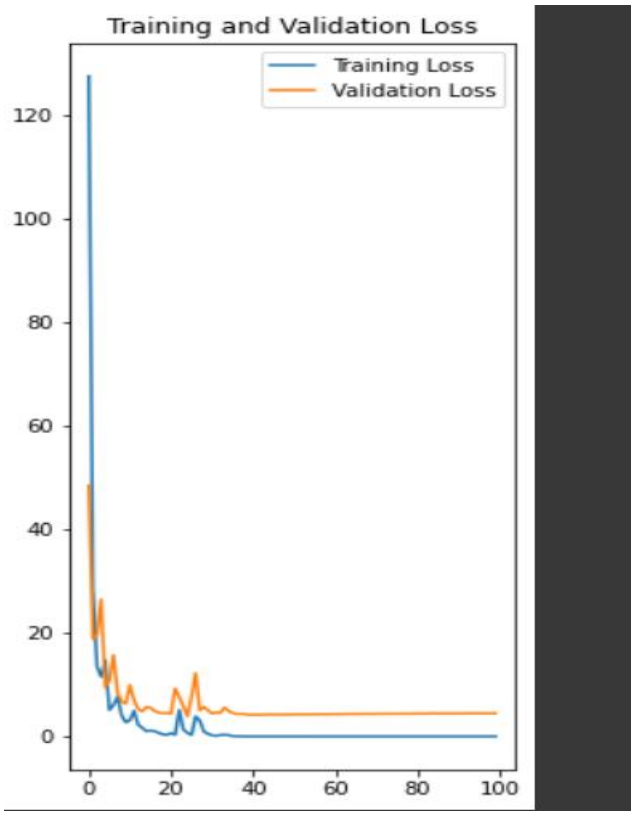
--> Artificial neural network (ANN) :

- Accuracy

```
test_loss, test_acc = model.evaluate(X_test, y_test, verbose=2)

test_acc_default = test_acc

print('\nTest accuracy:', round(test_acc*100, 2), "%")
```

```
4/4 - 0s - loss: 4.0609 - accuracy: 0.5607 - 34ms/epoch - 9ms/step

Test accuracy: 56.07 %
```

- loss curve



- Confusion Matrix && Confusion Matrix Curve