CSAI 801 Artificial Intelligence & Machine Learning W23.

Project: COVID-19 Outcome Prediction.

Supervised by:

Dr. Marwa Elsayed

T.A. Amr Zaki

T.A. Reem Abdel-salam

Name: Manar Samy Elghobashy

Id: 20398569

Contents

Introduction:

This project uses multiple classification algorithms to classify a dataset containing daily information on the number of affected cases, deaths, and recovery from the 2019 novel coronavirus. These different classifiers will be used to determine whether or not a person will recover from coronavirus symptoms based on certain pre-defined standard symptoms. These symptoms are based on World Health Organization recommendations (WHO). The data is available from 22 January 2020. Data is in the “data.csv” file.

Dataset view:

The dataset contains 863 records with 15 attributes. Attributes are 14 features and one label column. Columns are Unnamed: 0, country, location, age, gender, visited_wuhan, from_wuhan, symptoms, time_before_symptoms_appear, results.

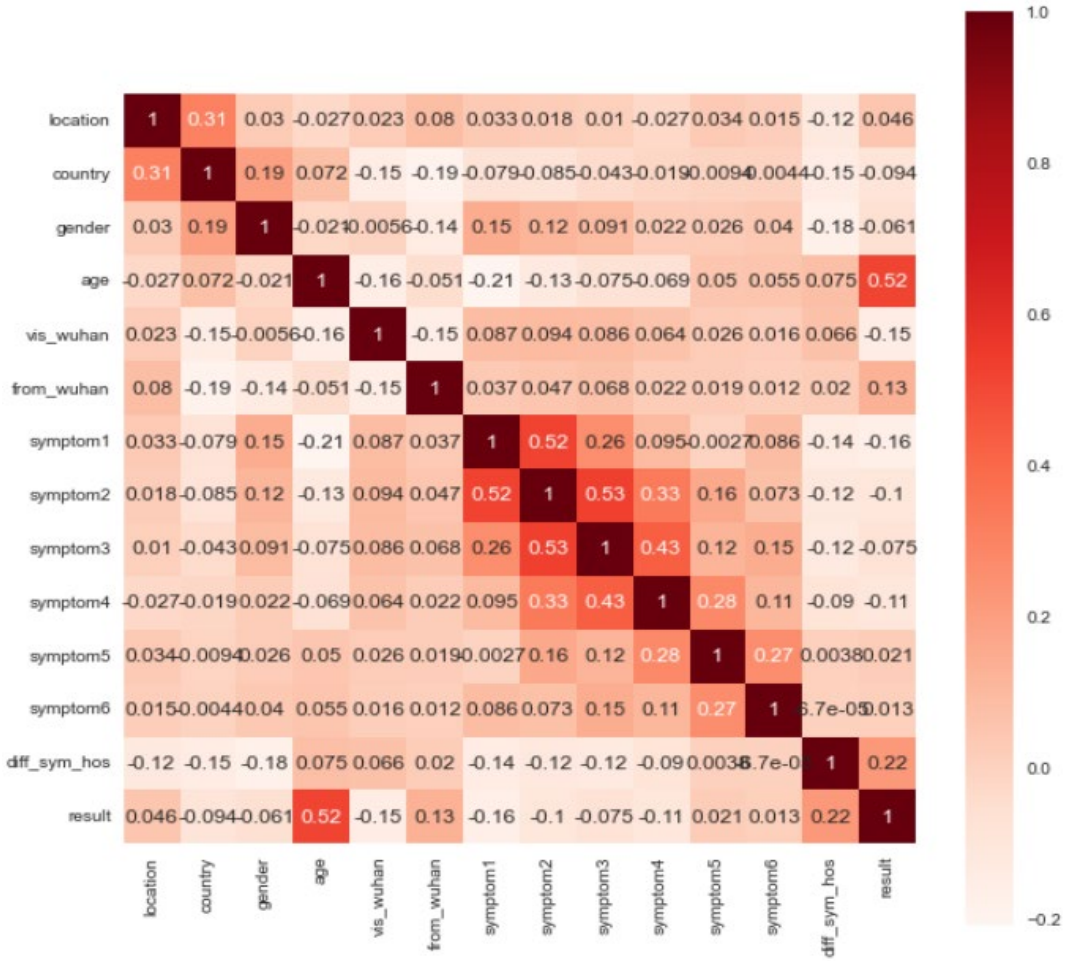
Unnamed: 0	location	country	gender	age	vis_wuhan	from_wuhan	symptom1	symptom2	symptom3	symptom4	symptom5	symptom6	diff_sym_hos	result
0	104	8	1	66.0	1	0	14	31	19	12	3	1	8	1
1	101	8	0	56.0	0	1	14	31	19	12	3	1	0	0
2	137	8	1	46.0	0	1	14	31	19	12	3	1	13	0
3	116	8	0	60.0	1	0	14	31	19	12	3	1	0	0
4	116	8	1	58.0	0	0	14	31	19	12	3	1	0	0

To know more about the data information, descriptions are displayed to discover more about its columns’ data types and their mean, standard deviation, etc.

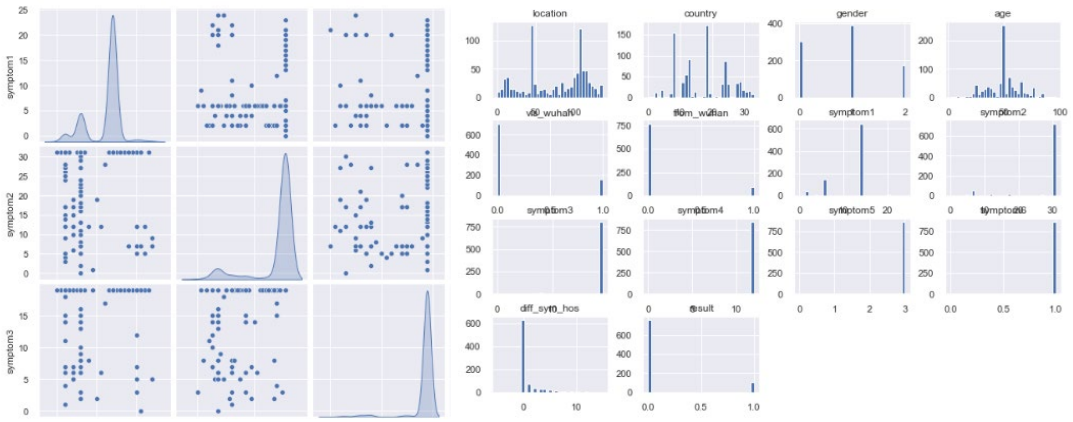
```
RangeIndex: 863 entries, 0 to 862
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      863 non-null    int64
 1   location        863 non-null    int64
 2   country         863 non-null    int64
 3   gender          863 non-null    int64
 4   age             863 non-null    float6
 5   vis_wuhan       863 non-null    int64
 6   from_wuhan      863 non-null    int64
 7   symptom1        863 non-null    int64
 8   symptom2        863 non-null    int64
 9   symptom3        863 non-null    int64
 10  symptom4        863 non-null    int64
 11  symptom5        863 non-null    int64
 12  symptom6        863 non-null    int64
 13  diff_sym_hos    863 non-null    int64
 14  result          863 non-null    int64
dtypes: float64(1), int64(14)
memory usage: 101.3 KB
```

	Unnamed: 0	location	country	gender	age	vis_wuhan	from_wuhan	symp
count	863.000000	863.000000	863.000000	863.000000	863.000000	863.000000	863.000000	863.
mean	431.000000	76.645423	16.995365	0.849363	49.400000	0.181924	0.107764	12.
std	249.270937	39.200264	7.809951	0.726062	15.079203	0.386005	0.310261	3.
min	0.000000	0.000000	0.000000	0.000000	2.000000	0.000000	0.000000	0.
25%	215.500000	45.000000	11.000000	0.000000	40.000000	0.000000	0.000000	14.
50%	431.000000	87.000000	18.000000	1.000000	49.400000	0.000000	0.000000	14.
75%	646.500000	110.000000	24.000000	1.000000	57.000000	0.000000	0.000000	14.
max	862.000000	138.000000	33.000000	2.000000	96.000000	1.000000	1.000000	24.

The data was already cleaned and preprocessed but needed some preparation before using it in building our models. So, in the beginning, the column ‘unnamed:0’ is dropped as it wasn’t a feature but an index column. Then a correlation map was designed to figure out the relation between these features.



And also, some visualization of the data was designed to know the distribution of each feature and explains the relations between the features.



The data was divided into features and labels containing 13 and 1 column, respectively. Then, the data was split into train and test by 70% for training and 30% for testing. Finally, the data was normalized using standard scalar to be convenient to the models.

Classification models:

All models are built using sklearn packages and had the same steps:

1. build the model with GridSearchCV to find optimal hyperparameters.
    - Used to tune hyperparameters.
    - Have a cross-validation parameter and it was set to '5' in all models.
    - An evaluation metric called scoring also was set to 'accuracy' in all models.
    - Parameters space to search on.
2. fit the model with the best hyperparameters.
3. compare the performance of all classifiers using different metrics such as the precision, recall, F1-score, and ROC/AUC curves.
4. calculating classification report.
5. confusion matrix, roc-auc, classification report visualization.

1. K-Nearest Neighbor:

KNN is a supervised, non-parametric model which is used to classify the data point based on similarity.

The parameter space used in the grid search:

- n_neighbors: from 1 to 15.
- weights: uniform, distance.

Best hyperparameter for KNN after using grid search:

- n_neighbors: 4.
- weights: distance.
- Best score: 0.9271763085399449

Fitting the model with the best hyperparameters:

- KNN Model Train Score is: 1.0

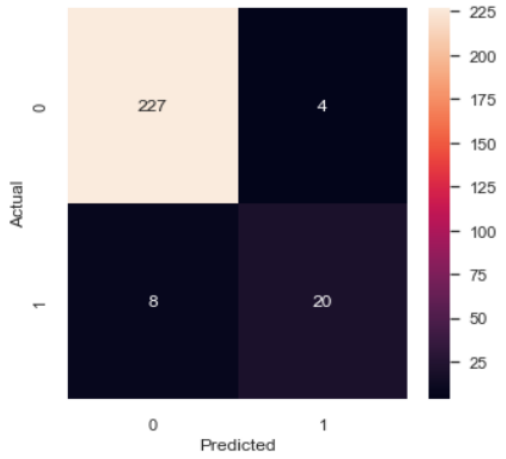
Metrics used to track the model performance:

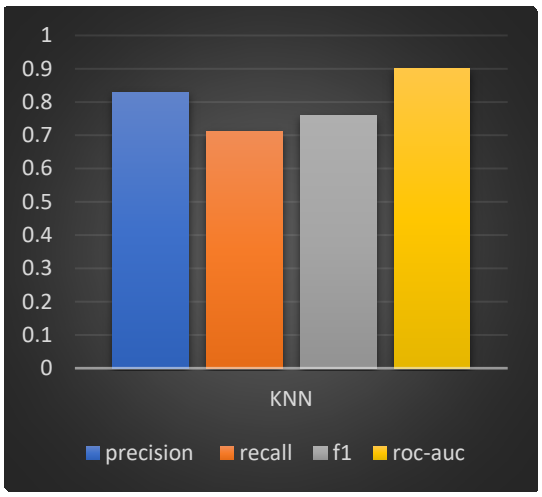
- confusion matrix for KNN is: [[227 4]
  [8 20]]
- accuracy score for KNN is: 0.9536679536679536
- The precision score for KNN is: 0.8333333333333334
- The recall score for KNN is: 0.7142857142857143
- The F1 score for KNN is: 0.7692307692307692
- The roc_auc score for KNN is: 0.9039888682745826

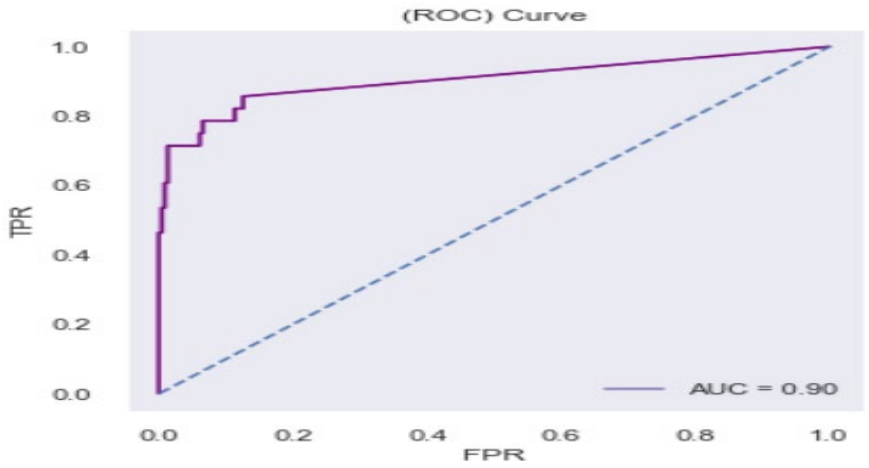
- Classification report:

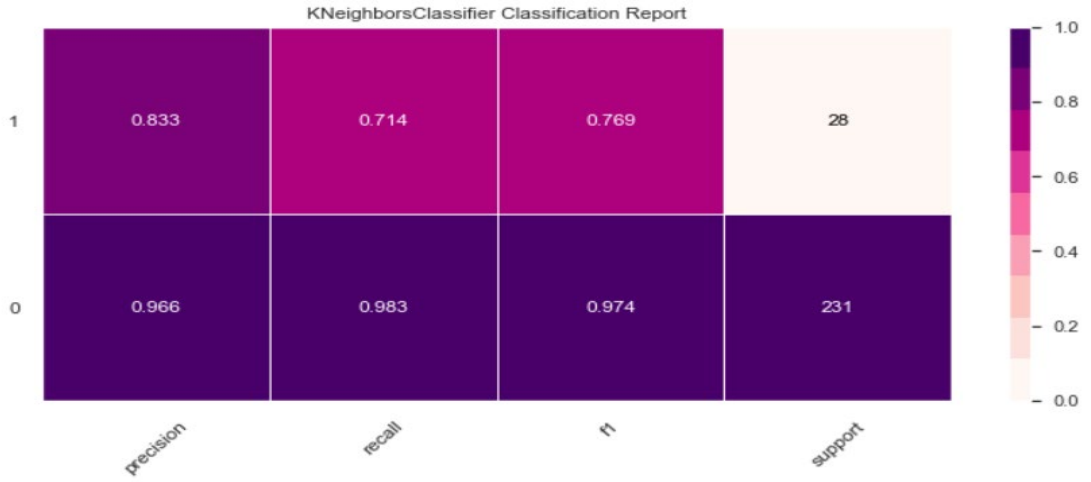
	Precision	Recall	F1	Support
0	0.97	0.98	0.97	231
1	0.83	0.71	0.77	28
Accuracy	---	---	0.95	259
Micro avg	0.90	0.85	0.87	259
Weighted avg	0.95	0.95	0.95	259

Visualizing model metrics:









2. Logistic Regression:

A supervised model is used to calculate or predict the probability of an event.

The parameter space used in the grid search:

- c: [0.001,0.005,0.08,0.1,0.5,0.8,1.2,1,5,10,25, 1e5].
- solver: ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'].

Best hyperparameter for KNN after using grid search:

- c: 0.5.
- solver: newton-cg.
- Best score: 0.9437327823691459.

Fitting the model with the best hyperparameters:

- Log reg Model Train Score is 0.9486754966887417.

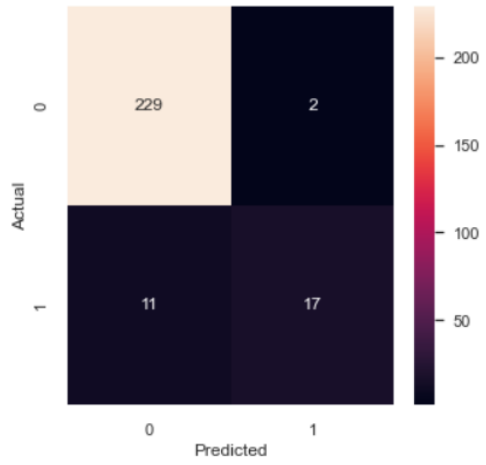
Metrics used to track the model performance:

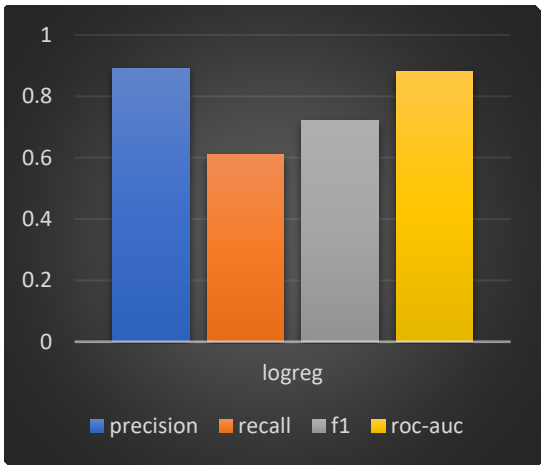
- confusion matrix for logistic regression is: [[229 2]
  [11 17]]
- accuracy score for logistic regression is: 0.9498069498069498
- The precision score for log reg is: 0.8947368421052632
- The recall score for log reg is: 0.6071428571428571
- The F1 score for log reg is: 0.7234042553191489
- The roc_auc score for log reg is: 0.8874458874458875

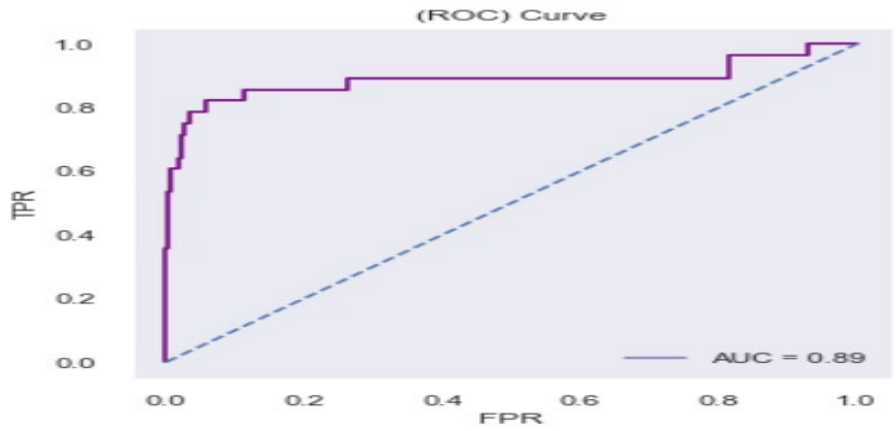
- Classification report:

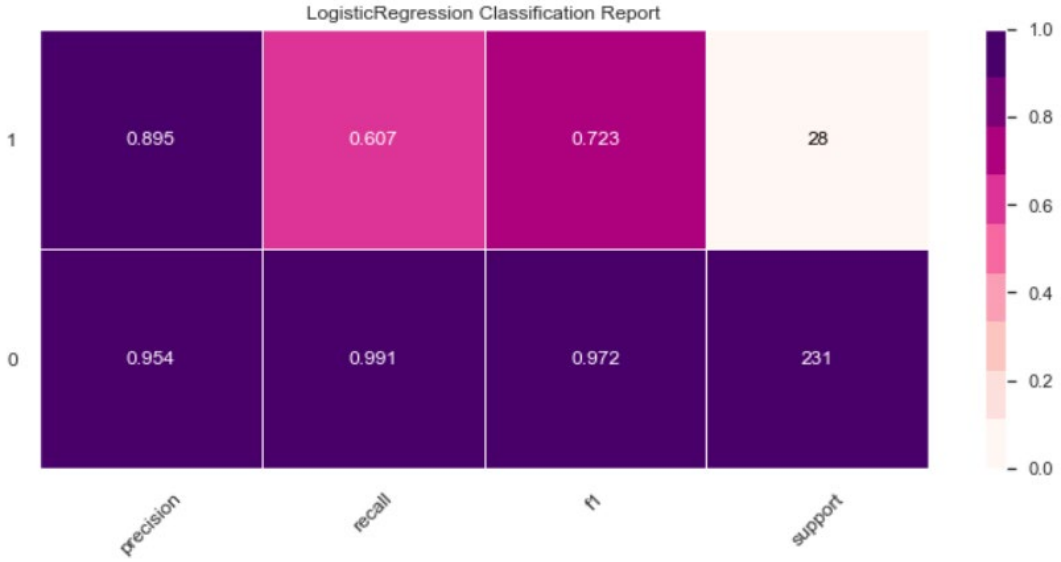
	Precision	Recall	F1	Support
0	0.975	0.99	0.97	231
1	0.89	0.61	0.72	28
Accuracy	---	---	0.95	259
Micro avg	0.92	0.80	0.85	259
Weighted avg	0.95	0.95	0.95	259

Visualizing model metrics:









3. Naïve Bayes:

A supervised model which applies Bayes' theorem with naïve assumption.

The parameter space used in the grid search:

- Var smoothing: (0, -10, num=500)

Best hyperparameter for KNN after using grid search:

- Var smoothing: 0.456371628192476.
- Best score: 0.8791735537190082

Fitting the model with the best hyperparameters:

- GaussianNB Model Train Score is 0.8741721854304636

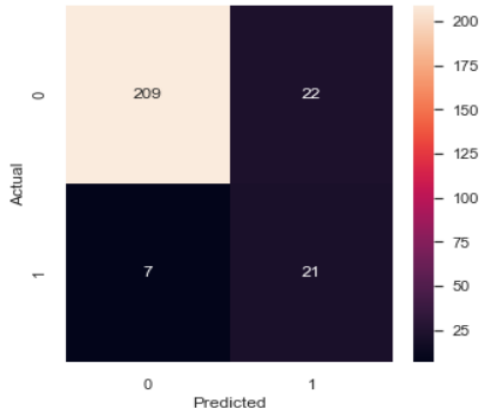
Metrics used to track the model performance:

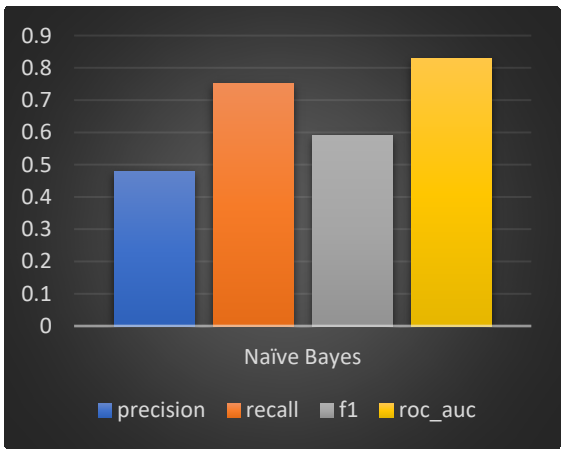
- confusion matrix for naïve bayes is: [[209 22]
  [7 21]]
- accuracy score for Naive Bayes is: 0.888030888030888
- the precision score for naïve Bayes is: 0.4883720930232558
- recall score for naïve Bayes is: 0.75
- F1 score for naïve Bayes is: 0.5915492957746479
- roc_auc score for naïve Bayes is: 0.839208410636982

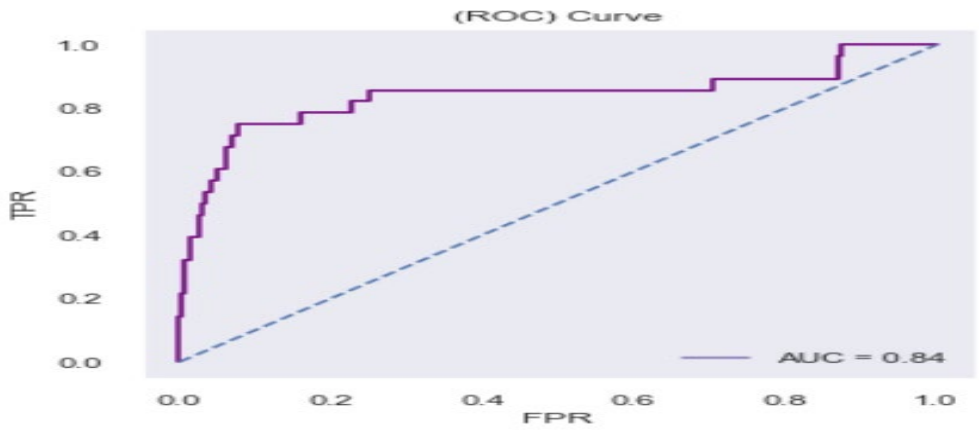
- Classification report:

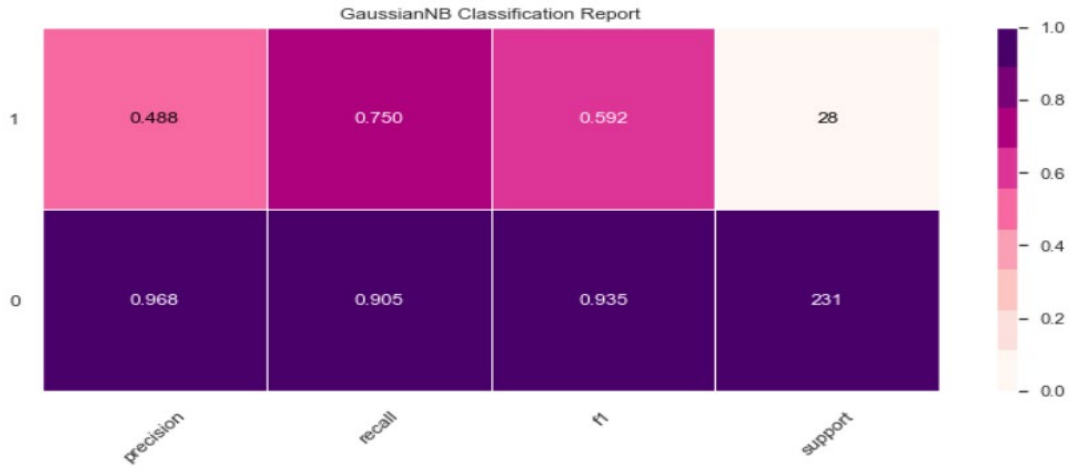
	Precision	Recall	F1	Support
0	0.97	0.90	0.94	231
1	0.49	0.75	0.59	28
Accuracy	---	---	0.89	259
Micro avg	0.73	0.83	0.76	259
Weighted avg	0.92	0.89	0.90	259

Visualizing model metrics:









4. Decision Tree:

a Supervised learning technique that can be used for both classification and Regression problems.

The parameter space used in the grid search:

- Criterion: ["gini", "entropy"]
- max_depth: from 1 to 15.
- min_samples_leaf: from 1 to 15.

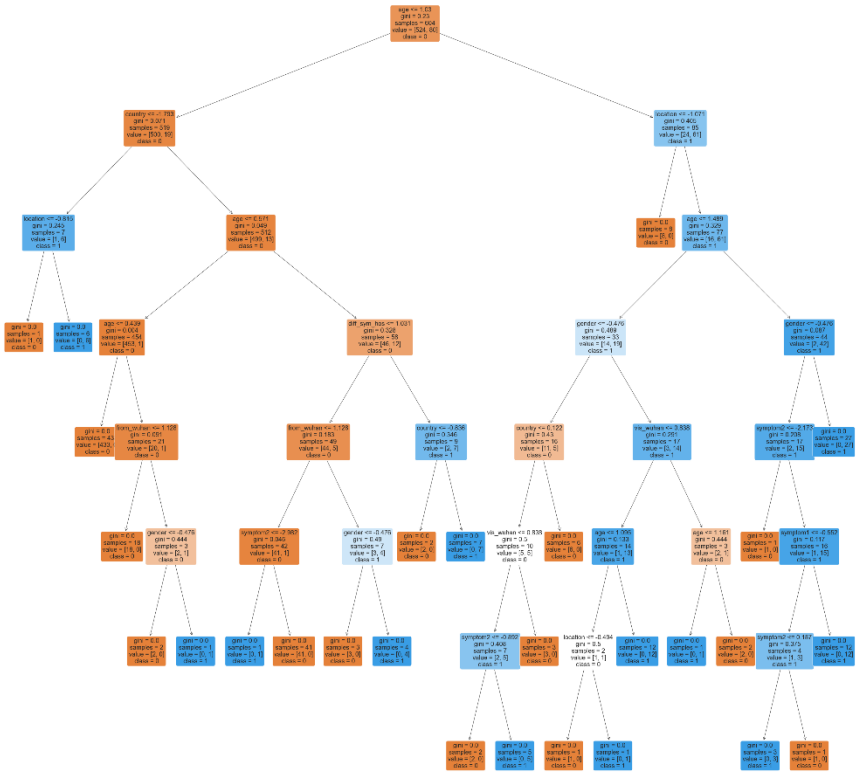
Best hyperparameter for Decision Tree after using grid search:

- Criterion: gini
- max_depth: 7.
- min_samples_leaf: 1.
- Best score: 0.9586225895316804.

Fitting the model with the best hyperparameters:

- Decision Tree Model Train Score is 1.0

Decision Tree graph:



Metrics used to track the model performance:

- confusion matrix for Decision Tree is: [[228 3]

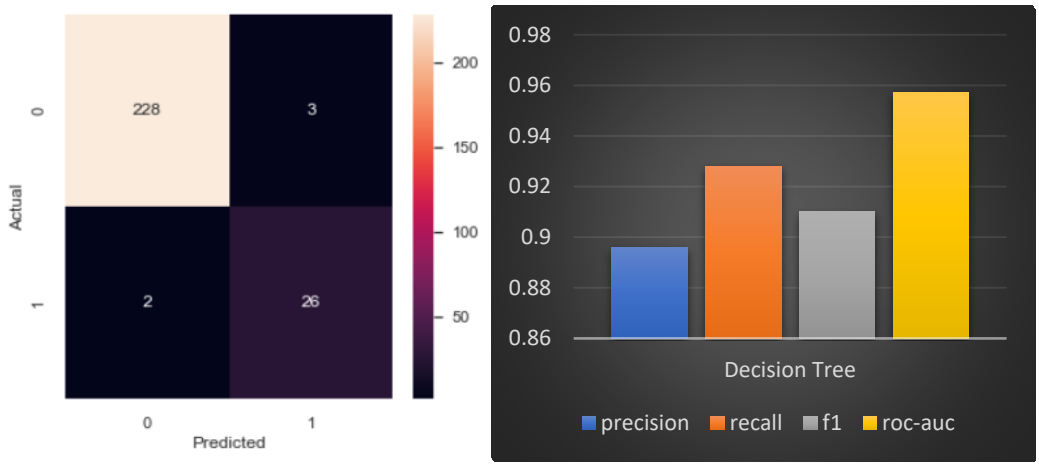
  [2 26]]

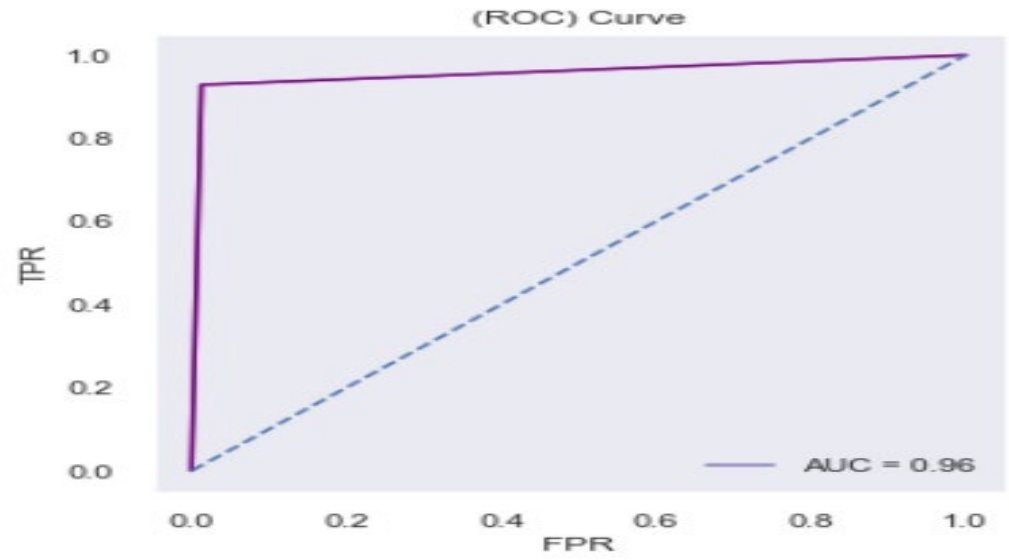
- accuracy score for Decision Tree is: 0.9806949806949807
- precision score for Decision Tree is: 0.896551724137931
- recall score for Decision Tree is: 0.9285714285714286
- F1 score for Decision Tree is: 0.912280701754386
- roc_auc score for Decision Tree is: 0.9577922077922079

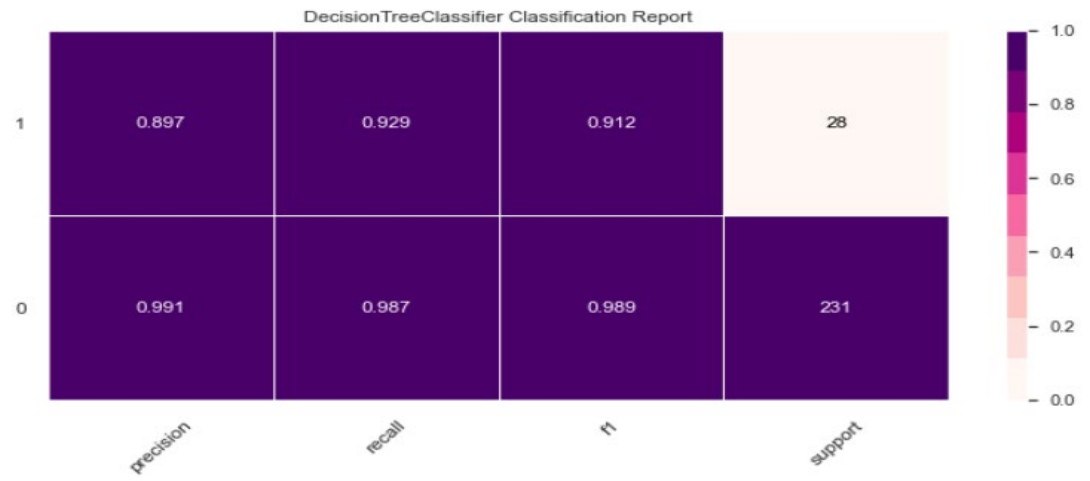
- Classification report:

	Precision	Recall	F1	Support
0	0.99	0.99	0.99	231
1	0.90	0.93	0.91	28
Accuracy	---	---	0.98	259
Micro avg	0.94	0.96	0.95	259
Weighted avg	0.98	0.98	0.98	259

Visualizing model metrics:







5. Support vector machine:

A supervised model uses classification algorithms for two-group classification problems.

The parameter space used in the grid search:

- C: [0.1, 1, 10, 100, 1000]
- gamma: [1, 0.1, 0.01, 0.001, 0.0001]
- kernel: 'rbf'

Best hyperparameter for support vector machine after using grid search:

- C: 10
- gamma: 0.1
- kernel: 'rbf'
- Best score: 0.9652341597796144

Fitting the model with the best hyperparameters:

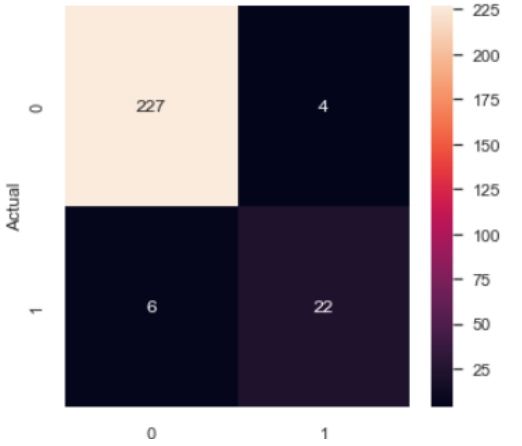
- SVM Model Train Score is 0.9966887417218543

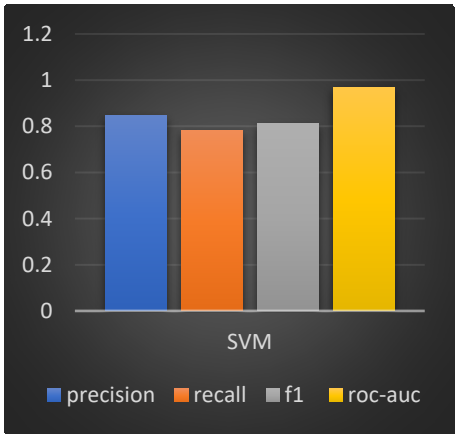
Metrics used to track the model performance:

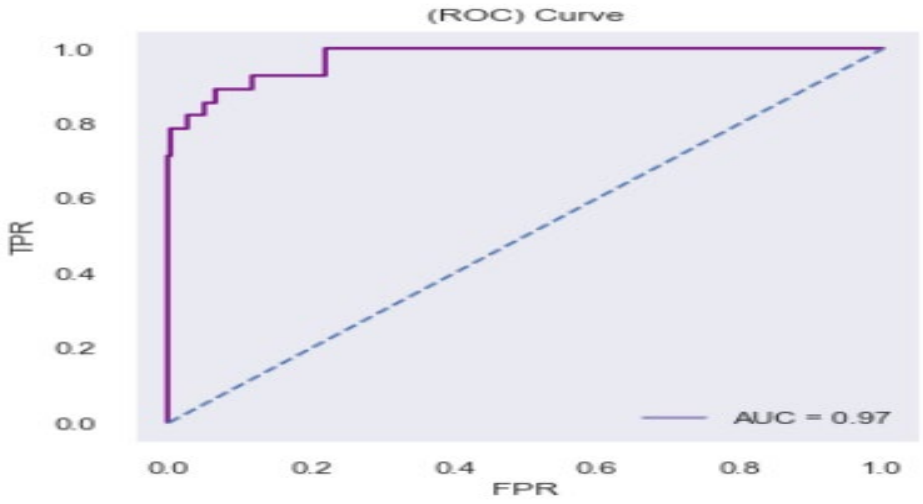
- confusion matrix for the support vector machine is: [[227 4]
  [6 22]]
- accuracy score for the support vector machine is: 0.896551724137931
- precision score for the support vector machine is: 0.8461538461538461
- the recall score for the support vector machine is: 0.7857142857142857
- F1 score for the support vector machine is: 0.8148148148148148
- roc_auc score for the support vector machine is: 0.9746444032158318
- Classification report:

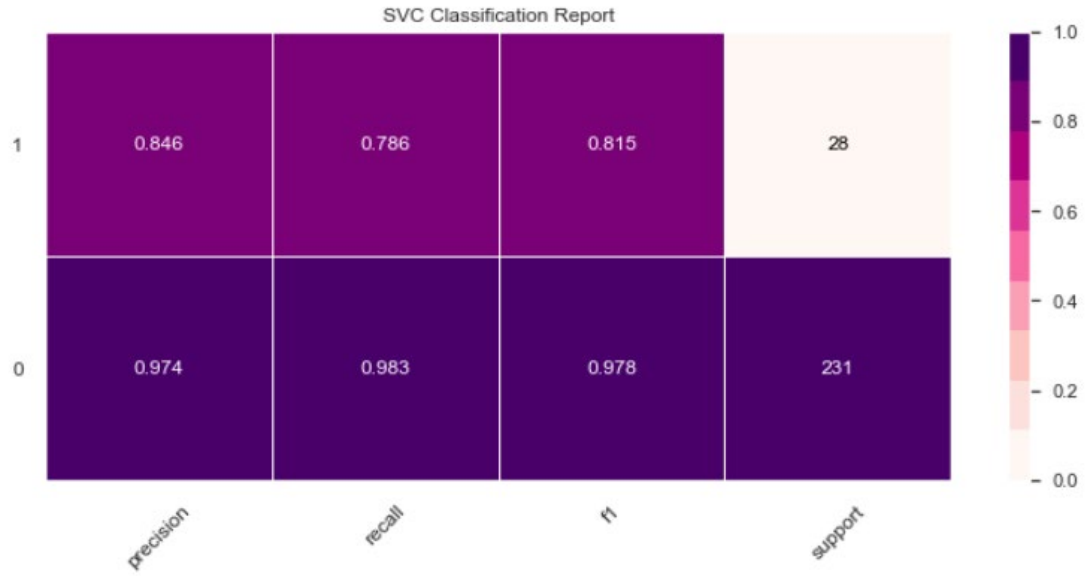
	Precision	Recall	F1	Support
0	0.97	0.98	0.98	231
1	0.85	0.79	0.81	28
Accuracy	---	---	0.96	259
Micro avg	0.91	0.88	0.90	259
Weighted avg	0.96	0.96	0.96	259

Visualizing model metrics:





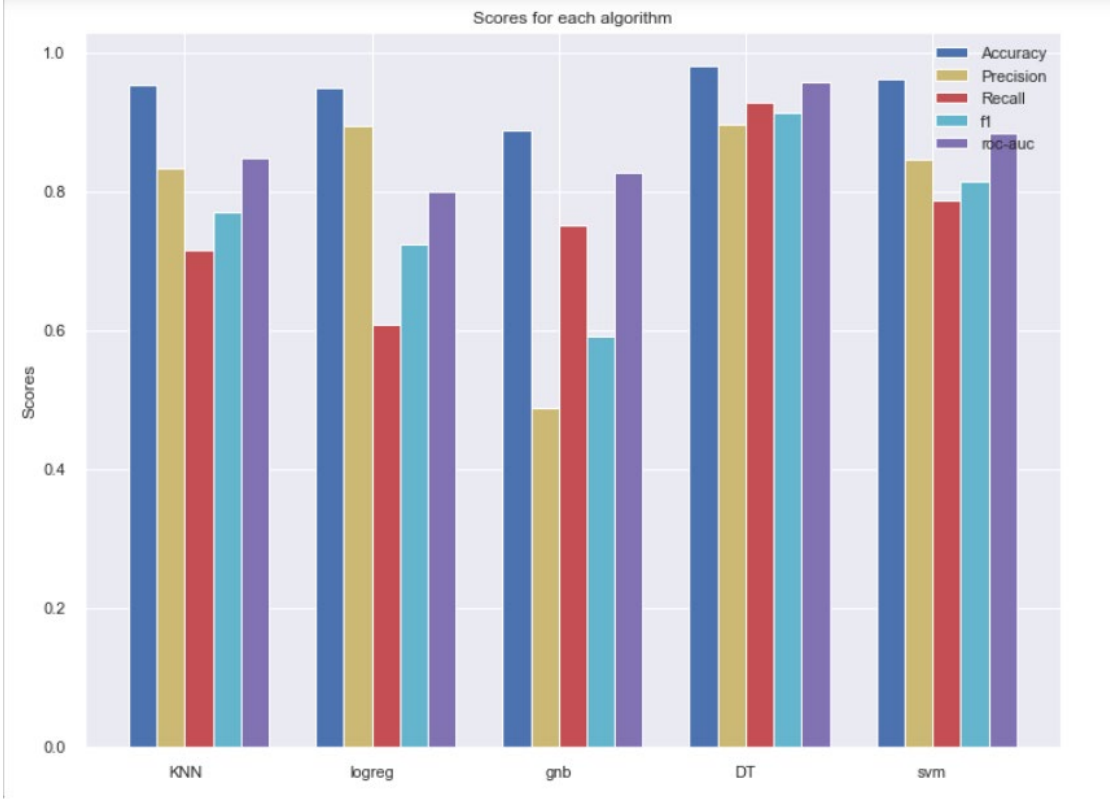




Comparison of models:

In the end, a table of scores and a graph were designed to compare the classification models and find the best model which fits the data perfectly.

	model	Recall	precision	f1	roc_auc	accracy
0	KNN	0.714286	0.833333	0.769231	0.848485	0.953668
1	logreg	0.607143	0.894737	0.723404	0.799242	0.949807
2	gnb	0.750000	0.488372	0.591549	0.827381	0.888031
3	DT	0.928571	0.896552	0.912281	0.957792	0.980695
4	svm	0.785714	0.846154	0.814815	0.884199	0.961390



From the previous observations, it is obvious that the Decision Tree model was the best for this problem as it has the highest scores. But as the data is split randomly the results may change.