

1. # Correlation between variables:

```
return model_table

[ ] # Baseline check:

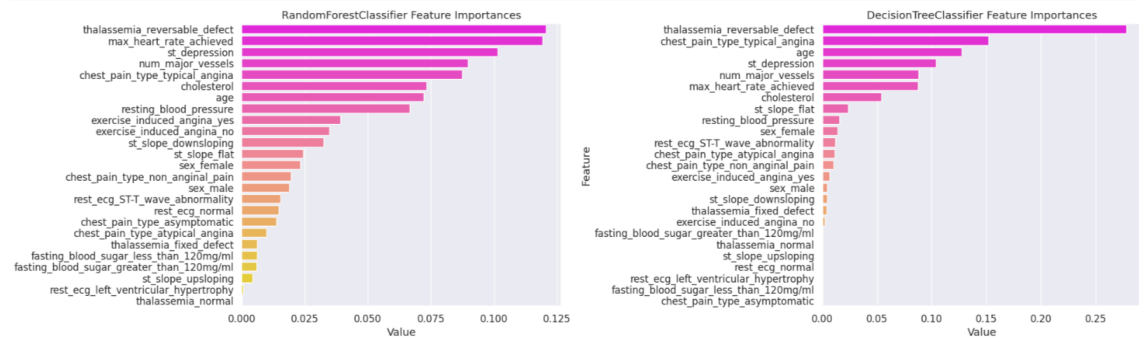
raw_models = model_check(X, y, classifiers, cv)

display(raw_models)
```

	Model Name	Train Roc/AUC Mean	Test Roc/AUC Mean	Test Roc/AUC Std	Train Accuracy Mean	Test Accuracy Mean	Test Acc Std	Train F1 Mean	Test F1 Mean	Test F1 Std	Time
3	RandomForestClassifier	1.000000	0.905593	0.029682	1.000000	0.831475	0.034684	1.000000	0.847947	0.027992	0.254913
4	GaussianNB	0.893298	0.858323	0.051020	0.815998	0.785464	0.045862	0.836326	0.807160	0.053222	0.005386
1	DecisionTreeClassifier	1.000000	0.776845	0.028428	1.000000	0.778798	0.023059	1.000000	0.799462	0.019614	0.005728
2	SVC	0.759426	0.731313	0.106884	0.685021	0.656448	0.067423	0.732524	0.721329	0.056617	0.009065
0	KNeighborsClassifier	0.845182	0.720570	0.051450	0.767306	0.673279	0.067152	0.791801	0.708466	0.070601	0.004202

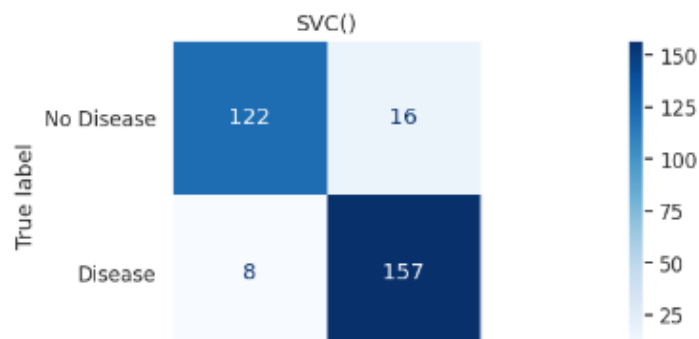
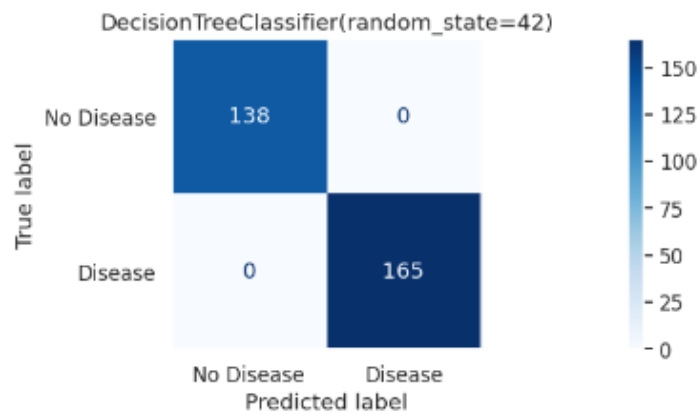
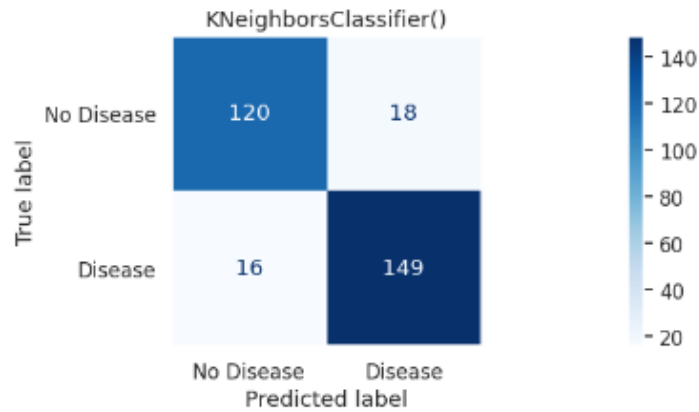
```
def f_imp(classifiers, X, y, bins):

    ''' A function for displaying important features'''
```



```
kbin_cat[col, X, nbins=5):

categorize = KBinDiscretizer(n_bins = nbins, encode = 'onehot', strategy = 'kmeans')
cat = categorize.fit_transform([col], values.reshape(-1,1))
cat = pd.DataFrame(cat, header=[])
```



```
[ ] print(y.shape)
    print(X_cat.shape)
```

```

# Print the best hyperparameters and corresponding scores for each classifier
print("Best hyperparameters for SVC: ", svc_grid.best_params_)
print("Best score for SVC: ", svc_grid.best_score_)
print("Best hyperparameters for Gaussian NB: ", gnb_grid.best_params_)
print("Best score for Gaussian NB: ", gnb_grid.best_score_)
print("Best hyperparameters for KNN: ", knn_grid.best_params_)
print("Best score for KNN: ", knn_grid.best_score_)
print("Best hyperparameters for Decision Tree: ", dtc_grid.best_params_)
print("Best score for Decision Tree: ", dtc_grid.best_score_)
print("Best hyperparameters for Random Forest: ", rfc_grid.best_params_)
print("Best score for Random Forest: ", rfc_grid.best_score_)

```

```

0.8688524590163934
0.7868852459016393
0.8360655737704918
0.8032786885245902
0.8852459016393442
Best hyperparameters for SVC: {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
Best score for SVC: 0.8263605442176871
Best hyperparameters for Gaussian NB: {'var_smoothing': 1e-05}
Best score for Gaussian NB: 0.7847789115646259
Best hyperparameters for KNN: {'n_neighbors': 5, 'p': 2}
Best score for KNN: 0.8139455782312925
Best hyperparameters for Decision Tree: {'criterion': 'entropy', 'max_depth': 4, 'min_samples_split': 6}
Best score for Decision Tree: 0.7892006802721088
Best hyperparameters for Random Forest: {'max_depth': 4, 'min_samples_split': 2, 'n_estimators': 200}
Best score for Random Forest: 0.8427721088435373

```