

TEAM ID	PNT2022TMID04216
PROJECT NAME	Statistical Machine Learning Approaches to Liver Disease Prediction

Team Leader: Vigneshwar M

Team Member: Mohanakrishnan R

Team Member: Dhesheswar K

Team Member: Nitish Kumar R

MODEL Evaluation

Finally, we need to check to see how well our model is performing on the test data. Evaluation Metrics:

accuracy_score of SVM is

```
# Checking for accuracy score from actual data and predicted data
SVMaccuracy=accuracy_score(SVMpred, ytest)
SVMaccuracy

0.7606837606837606
```

accuracy_score of Random forest classification is

```
#Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()

# train the data with Random Forest model
RFmodel.fit(xtrain, ytrain)

RandomForestClassifier()

RFpred=RFmodel.predict(xtest)

# Checking for accuracy score from actual data and predicted data
RFaccuracy=accuracy_score(RFpred, ytest)
RFaccuracy

0.7094017094017094

# showing the confusion matrix
RFcm=confusion_matrix(RFpred, ytest)
RFcm

array([[77, 22],
       [12,  6]], dtype=int64)
```

Model Evaluation

Finally, we need to check to see how well our model is performing on the test data. Evaluation Metrics:

accuracy_score of SVM is

```
# Checking for accuracy score from actual data and predicted data
SVMaccuracy=accuracy_score(SVMpred, ytest)
SVMaccuracy

0.7606837606837606
```

accuracy_score of Random forest classification is

```
#Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()

# train the data with Random Forest model
RFmodel.fit(xtrain, ytrain)

RandomForestClassifier()

RFpred=RFmodel.predict(xtest)

# Checking for accuracy score from actual data and predicted data
RFaccuracy=accuracy_score(RFpred, ytest)
RFaccuracy

0.7094017094017094

# showing the confusion matrix
RFcm=confusion_matrix(RFpred, ytest)
RFcm

array([[77, 22],
       [12,  6]], dtype=int64)
```

As we can see that the accuracy_score of the Support vector machine is higher compare to KNN and Random forest algorithms, we are proceeding with the support vector machine model.

Model Evaluation

```
In [4]: model = ensemble.RandomForestClassifier()
model.fit(X_train, y_train) #Put X_Train.values while running The app.py---
y_pred = model.predict(X_test)
print('Accuracy : {}'.format(accuracy_score(y_test, y_pred)))

clf_report = classification_report(y_test, y_pred)
print('Classification report')
print("-----")
print(clf_report)
print("-----")
```

Accuracy : 1.0

Classification report

	precision	recall	f1-score	support
1	1.00	1.00	1.00	2038
2	1.00	1.00	1.00	836
accuracy			1.00	2874
macro avg	1.00	1.00	1.00	2874
weighted avg	1.00	1.00	1.00	2874