

Project Development Phase

Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID15358
Project Name	STATISTICAL MACHINE LEARNING APPROACHES TO LIVER DISEASE PREDICTION
Maximum Marks	10 Marks

Model Performance Test

S.No.	Parameter	Values	Screenshot
1.	Metrics	Classification Model: Accuracy Score- & Classification Report	Random Forest Algorithm, Gradient Boost Algorithm, AdaBoost Algorithm
2.	Tune the model	Hyperparameter Tuning method	Confusion Matrix

The screenshots are provided below for the above-mentioned table.

1. Metrics

Model: Random Forest Algorithm

There is no need of Standardization and Normalization of our dataset, as we using Ensemble Technique.

```
# Importing Performance Metrics:
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
# RandomForestClassifier:
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier()
RandomForest = RandomForest.fit(X_train,y_train)
```

```
# Predictions:
y_pred = RandomForest.predict(X_test)
```

```
# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Accuracy: 0.8516949152542372

[[94 20]

[15 107]]

	precision	recall	f1-score	support
1	0.86	0.82	0.84	114
2	0.84	0.88	0.86	122
accuracy			0.85	236
macro avg	0.85	0.85	0.85	236
weighted avg	0.85	0.85	0.85	236

GradientBoost Algorithm

```
# GradientBoostingClassifier:
from sklearn.ensemble import GradientBoostingClassifier
GradientBoost = GradientBoostingClassifier()
GradientBoost = GradientBoost.fit(X_train,y_train)
```

```
# Predictions:
y_pred = GradientBoost.predict(X_test)
```

```
# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Accuracy: 0.8220338983050848

[[91 23]

[19 103]]

	precision	recall	f1-score	support
1	0.83	0.80	0.81	114
2	0.82	0.84	0.83	122
accuracy			0.82	236
macro avg	0.82	0.82	0.82	236
weighted avg	0.82	0.82	0.82	236

AdaBoost Algorithm

```
# AdaBoostClassifier:
from sklearn.ensemble import AdaBoostClassifier
AdaBoost = AdaBoostClassifier()
AdaBoost = AdaBoost.fit(X_train,y_train)

# Predictions:
y_pred = AdaBoost.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Accuracy: 0.7457627118644068

[[88 26]

[34 88]]

	precision	recall	f1-score	support
1	0.72	0.77	0.75	114
2	0.77	0.72	0.75	122
accuracy			0.75	236
macro avg	0.75	0.75	0.75	236
weighted avg	0.75	0.75	0.75	236

2. Tune the model

Confusion Matrix

```
# Importing Performance Metrics:
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
# RandomForestClassifier:
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier()
RandomForest = RandomForest.fit(X_train,y_train)

# Predictions:
y_pred = RandomForest.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```