Model Development Phase

Date	03-10-2024
Team ID	LTVIP2024TMID24892
Project Title	Liver Patient Identification – prediction of liver patient
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in this. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

Train Test split for all models:

```
# Train Test Split:
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X_smote,y_smote, test_size=0.2, random_state=42)
```

Random forest model:

```
# 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(classification_report(y_test,y_pred))
```

KNN Model:

```
#kNearest Neighbours
from sklearn.neighbors import KNeighborsClassifier
kneighbours = KNeighborsClassifier()
kneighbours = kneighbours.fit(X_train,y_train)

# Predictions:
y_pred = kneighbours.predict(X_test)

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

Support vector Machine Model:

```
#kNearest Neighbours
from sklearn.neighbors import KNeighborsClassifier
kneighbours = KNeighborsClassifier()
kneighbours = kneighbours.fit(X_train,y_train)

# Predictions:
y_pred = kneighbours.predict(X_test)

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

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	`print(classification_report(y_test,y_pred))` Accuracy: 0.8291139240506329 precision recall f1-score support 0 0.91 0.74 0.82 82 2 0.77 0.92 0.84 76 accuracy 0.83 158 macro avg 0.84 0.83 0.83 158 weighted avg 0.84 0.83 0.83 158	82%	`confusion_matrix(y_test,y_pred)` array([[61, 21], [6, 70]])

Support Vector Machine	`print(classification_report(y_test,y_pred))` Accuracy: 0.7531645569620253 precision recall f1-score support 0 0.86 0.62 0.72 82 2 0.69 0.89 0.78 76 accuracy 0.75 158 macro avg 0.78 0.76 0.75 158 weighted avg 0.78 0.75 0.75 158	75%	`confusion_matrix(y_test,y_pred)` array([[40, 42], [9, 67]])
KNN	`print(classification_report(y_test,y_pred))` Accuracy: 0.6772151898734177 precision recall f1-score support 0 0.82 0.49 0.61 82 2 0.61 0.88 0.72 76 accuracy 0.68 158 macro avg 0.72 0.68 0.67 158 weighted avg 0.72 0.68 0.67 158	67%	`print(confusion_matrix(y_test,y_pred))` [[51 31] [8 68]]