

Model Development Phase Template

Date	15 july 2024
Team ID	739952
Project Title	Prediction and Analysis of Liver Patient Data Using Machine Learning
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. 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:

Logistic Regression

```
# LogisticRegression  
from sklearn.linear_model import LogisticRegression  
lr = LogisticRegression()  
lr.fit(x_train, y_train)  
y_pred_lr = lr.predict(x_test)  
y_pred_lr
```

KNeighborsClassifier

```
#KNeighborsClassifier  
from sklearn.neighbors import KNeighborsClassifier  
knn = KNeighborsClassifier()  
knn.fit(x_train, y_train)  
ypred_knn = knn.predict(x_test)
```

SVC

```
#SVC()
from sklearn.svm import SVC
svm = SVC()
svm.fit(x_train, y_train)
y_pred_svm = svm.predict(x_test)
```

RandomForestClassifier

```
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()
rfc.fit(x_train, y_train)
ypred_rfc = rfc.predict(x_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Logistic Regression	<pre>print(classification_report(y_test,y_pred_lr))</pre> <pre> precision recall F1-score support 1 0.75 0.50 0.63 128 2 0.45 0.15 0.27 47 accuracy_ 0.60 0.55 0.72 175 macro avg_ 0.60 0.55 0.68 175 weighted avg_ 0.67 0.72 0.68 175</pre>	<pre>lr_acc = accuracy_score(y_pred, y_test) lr_acc</pre> <pre>0.72</pre>	<pre>confusion_matrix(y_test,y_pred_lr) print(confmat)</pre> <pre>[[117 11] [38 9]]</pre>
K neighbors Classifier	<pre>print(classification_report(y_test,ypred_knn))</pre> <pre> precision recall F1-score support 1 0.51 0.88 0.66 100 2 0.42 0.43 0.43 37 accuracy_ 0.51 0.62 0.71 137 macro avg_ 0.51 0.62 0.61 137 weighted avg_ 0.71 0.71 0.71 137</pre>	<pre>knn_acc = accuracy_score(ypred_knn, y_test) print(knn_acc)</pre> <pre>0.7080292043795624</pre>	<pre>confusion_matrix(y_test,ypred_knn)</pre> <pre>array([[87, 22], [21, 16]], dtype=int64)</pre>

Random Forest Classifier

```
print(classification_report(y_test,y_pred_rfc))
```

	precision	recall	f1-score	support
1	0.80	0.85	0.82	87
2	0.46	0.57	0.41	50
accuracy			0.73	137
macro avg	0.63	0.71	0.61	137
weighted avg	0.71	0.73	0.72	137

```
f1_score = accuracy_score(y_pred_rf, y_test)
print(f1_score)
0.7264577264577265
```

```
confusion_matrix(y_test,y_pred_rfc)
```

```
array([[74, 13],
       [19, 11]], dtype=int64)
```

SVC

```
print(classification_report(y_test,y_pred_svc))
```

	precision	recall	f1-score	support
1	0.74	1.00	0.85	87
2	0.00	0.00	0.00	50
accuracy			0.74	137
macro avg	0.37	0.50	0.43	137
weighted avg	0.55	0.34	0.43	137

```
accuracy_score(y_pred_svc, y_test)
0.7435839743583974
```

```
confusion_matrix(y_test,y_pred_svc)
```

```
array([[87,  0],
       [30,  0]], dtype=int64)
```