

Model Development Phase Template

Date	09 July 2024
Team ID	SWTID1720023141
Project Title	Prediction and Analysis of Liver Patient Data Using Machine Learning
Maximum Marks	6 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 Selection Report:

Model	Classification Report	Accuracy	Confusion Matrix																													
Logistic Regression	<pre>print(classification_report(y_test,y_pred))</pre>																															
	<table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1</td><td>0.75</td><td>0.91</td><td>0.83</td><td>128</td></tr><tr><td>2</td><td>0.45</td><td>0.19</td><td>0.27</td><td>47</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.72</td><td>175</td></tr><tr><td>macro avg</td><td>0.60</td><td>0.55</td><td>0.55</td><td>175</td></tr><tr><td>weighted avg</td><td>0.67</td><td>0.72</td><td>0.68</td><td>175</td></tr></tbody></table>		precision	recall	f1-score	support	1	0.75	0.91	0.83	128	2	0.45	0.19	0.27	47	accuracy			0.72	175	macro avg	0.60	0.55	0.55	175	weighted avg	0.67	0.72	0.68	175	<pre>lr_acc = accuracy_score(y_pred, y_test)</pre> <pre>lr_acc</pre> 0.72
	precision	recall	f1-score	support																												
1	0.75	0.91	0.83	128																												
2	0.45	0.19	0.27	47																												
accuracy			0.72	175																												
macro avg	0.60	0.55	0.55	175																												
weighted avg	0.67	0.72	0.68	175																												

K neighbors Classifier	<pre>print(classification_report(y_test,ypred_knn))</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1</td><td>0.81</td><td>0.80</td><td>0.80</td><td>109</td></tr><tr><td>2</td><td>0.42</td><td>0.43</td><td>0.43</td><td>37</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.71</td><td>146</td></tr><tr><td>macro avg</td><td>0.61</td><td>0.62</td><td>0.61</td><td>146</td></tr><tr><td>weighted avg</td><td>0.71</td><td>0.71</td><td>0.71</td><td>146</td></tr></tbody></table>		precision	recall	f1-score	support	1	0.81	0.80	0.80	109	2	0.42	0.43	0.43	37	accuracy			0.71	146	macro avg	0.61	0.62	0.61	146	weighted avg	0.71	0.71	0.71	146	<pre>knn_acc = accuracy_score(ypred_knn, y_test) print(knn_acc)</pre> <p>0.7054794520547946</p>	<pre>confusion_matrix(y_test,ypred_knn)</pre> <p>array([[87, 22], [21, 16]], dtype=int64)</p>
	precision	recall	f1-score	support																													
1	0.81	0.80	0.80	109																													
2	0.42	0.43	0.43	37																													
accuracy			0.71	146																													
macro avg	0.61	0.62	0.61	146																													
weighted avg	0.71	0.71	0.71	146																													
Random Forest Classifier	<pre>print(classification_report(y_test,ypred_rfc))</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1</td><td>0.80</td><td>0.85</td><td>0.82</td><td>87</td></tr><tr><td>2</td><td>0.46</td><td>0.37</td><td>0.41</td><td>30</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.73</td><td>117</td></tr><tr><td>macro avg</td><td>0.63</td><td>0.61</td><td>0.61</td><td>117</td></tr><tr><td>weighted avg</td><td>0.71</td><td>0.73</td><td>0.72</td><td>117</td></tr></tbody></table>		precision	recall	f1-score	support	1	0.80	0.85	0.82	87	2	0.46	0.37	0.41	30	accuracy			0.73	117	macro avg	0.63	0.61	0.61	117	weighted avg	0.71	0.73	0.72	117	<pre>rfc_acc = accuracy_score(ypred_rfc, y_test) print(rfc_acc)</pre> <p>0.7264957264957265</p>	<pre>confusion_matrix(y_test,ypred_rfc)</pre> <p>array([[74, 13], [19, 11]], dtype=int64)</p>
	precision	recall	f1-score	support																													
1	0.80	0.85	0.82	87																													
2	0.46	0.37	0.41	30																													
accuracy			0.73	117																													
macro avg	0.63	0.61	0.61	117																													
weighted avg	0.71	0.73	0.72	117																													
SVC	<pre>print(classification_report(y_test,y_pred_svm))</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1</td><td>0.74</td><td>1.00</td><td>0.85</td><td>87</td></tr><tr><td>2</td><td>0.00</td><td>0.00</td><td>0.00</td><td>30</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.74</td><td>117</td></tr><tr><td>macro avg</td><td>0.37</td><td>0.50</td><td>0.43</td><td>117</td></tr><tr><td>weighted avg</td><td>0.55</td><td>0.74</td><td>0.63</td><td>117</td></tr></tbody></table>		precision	recall	f1-score	support	1	0.74	1.00	0.85	87	2	0.00	0.00	0.00	30	accuracy			0.74	117	macro avg	0.37	0.50	0.43	117	weighted avg	0.55	0.74	0.63	117	<pre>accuracy_score(y_pred_svm, y_test)</pre> <p>0.7435897435897436</p>	<pre>confusion_matrix(y_test,y_pred_svm)</pre> <p>array([[87, 0], [30, 0]], dtype=int64)</p>
	precision	recall	f1-score	support																													
1	0.74	1.00	0.85	87																													
2	0.00	0.00	0.00	30																													
accuracy			0.74	117																													
macro avg	0.37	0.50	0.43	117																													
weighted avg	0.55	0.74	0.63	117																													