

## Model Development Phase Template

Date	15 July 2024
Team ID	739961
Project Title	SmartLender - Automotive Kickstart
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:

```
J: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import r2_score
lr = LogisticRegression()
lr.fit(x_train,y_train)
```

```
pred=lr.predict(x_test)
pred
```

```
array([1, 1, 1, ..., 3, 1, 1])
```

```
lg_ac=accuracy_score(y_test,pred)
lg_f1 = f1_score(y_test, pred, average='weighted') # Options: 'micro', 'macro', 'weighted'
lg_r2=r2_score(y_test,pred)
print(lg_ac)
print(lg_f1)
print(lg_r2)
```

```
0.8042576462898987
```

```
0.7560252728477291
```

```
0.5704852951733073
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import r2_score
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
```

```
kpred=knn.predict(x_test)
kpred
```

```
array([1, 3, 1, ..., 3, 3, 1])
```

```
knn_ac=accuracy_score(y_test,kpred)
knn_f1 = f1_score(y_test, kpred, average='weighted') # Options: 'micro', 'macro', 'weighted'
knn_r2=r2_score(y_test,kpred)
print(knn_ac)
print(knn_f1)
print(knn_r2)
```

```
0.827373250990383
0.7963496160215812
0.6793182337664017
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import r2_score
rf=RandomForestClassifier()
rf.fit(x_train, y_train)
```

```
rpred=rf.predict(x_test)
rpred
```

```
array([1, 1, 1, ..., 3, 3, 1])
```

```
rf_ac=accuracy_score(y_test, rpred)
rf_f1 = f1_score(y_test, rpred, average='weighted') # Options: 'micro', 'macro', 'weighted'
rf_r2=r2_score(y_test, rpred)
print(rf_ac)
print(rf_f1)
print(rf_r2)
```

```
0.8595590294914034
0.8253275461008338
0.7824806763876369
```

```
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import r2_score
linear_svc = LinearSVC()
linear_svc.fit(x_train, y_train)
```

```
spred=rf.predict(x_test)
spred
```

```
array([1, 1, 1, ..., 3, 3, 1])
```

```
svm_ac=accuracy_score(y_test, spred)
svm_f1 = f1_score(y_test, spred, average='weighted') # Options: 'micro', 'macro', 'weighted'
svm_r2=r2_score(y_test, rpred)
print(svm_ac)
print(svm_f1)
print(svm_r2)
```

```
0.8595590294914034
0.8253275461008338
0.7824806763876369
```

## Model Validation and Evaluation Report:

MODEL	Classification Report	F1-score	Accuracy Score
Logistic Regression	<pre>print('\n\n', classification_report(y_test,pred))</pre> <pre>               precision    recall  f1-score   support      0       0.00      0.00      0.00       7766     1       0.75      0.95      0.84      39471     2       1.00      0.01      0.02        552     3       0.91      0.84      0.87      26831     4       0.00      0.00      0.00        351   accuracy          0.80      74971  macro avg       0.53      0.36      0.35      74971  weighted avg    0.73      0.80      0.76      74971 </pre>	75.60 %	80.43%
RandomForest	<pre>print('\n\n', classification_report(y_test,kpred))</pre> <pre>               precision    recall  f1-score   support      0       0.22      0.07      0.11       7766     1       0.81      0.91      0.86      39471     2       0.12      0.01      0.01        552     3       0.92      0.95      0.93      26831     4       0.00      0.00      0.00        351   accuracy          0.83      74971  macro avg       0.41      0.39      0.38      74971  weighted avg    0.78      0.83      0.80      74971 </pre>	85.96 %	82.53%

KNN	<pre>print('\n\n\n', classification_report(y_test,rpred))</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.28</td><td>0.07</td><td>0.12</td><td>7766</td></tr><tr><td>1</td><td>0.83</td><td>0.94</td><td>0.88</td><td>39471</td></tr><tr><td>2</td><td>0.26</td><td>0.03</td><td>0.06</td><td>552</td></tr><tr><td>3</td><td>0.95</td><td>0.99</td><td>0.97</td><td>26831</td></tr><tr><td>4</td><td>0.10</td><td>0.01</td><td>0.03</td><td>351</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>74971</td></tr><tr><td>macro avg</td><td>0.49</td><td>0.41</td><td>0.41</td><td>74971</td></tr><tr><td>weighted avg</td><td>0.81</td><td>0.86</td><td>0.83</td><td>74971</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.28	0.07	0.12	7766	1	0.83	0.94	0.88	39471	2	0.26	0.03	0.06	552	3	0.95	0.99	0.97	26831	4	0.10	0.01	0.03	351	accuracy			0.86	74971	macro avg	0.49	0.41	0.41	74971	weighted avg	0.81	0.86	0.83	74971	82.74 %	79.63%
	precision	recall	f1-score	support																																												
0	0.28	0.07	0.12	7766																																												
1	0.83	0.94	0.88	39471																																												
2	0.26	0.03	0.06	552																																												
3	0.95	0.99	0.97	26831																																												
4	0.10	0.01	0.03	351																																												
accuracy			0.86	74971																																												
macro avg	0.49	0.41	0.41	74971																																												
weighted avg	0.81	0.86	0.83	74971																																												
SVM	<pre>print('\n\n\n', classification_report(y_test,spred))</pre> <div>ogle output scrolling</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.28</td><td>0.07</td><td>0.12</td><td>7766</td></tr><tr><td>1</td><td>0.83</td><td>0.94</td><td>0.88</td><td>39471</td></tr><tr><td>2</td><td>0.26</td><td>0.03</td><td>0.06</td><td>552</td></tr><tr><td>3</td><td>0.95</td><td>0.99</td><td>0.97</td><td>26831</td></tr><tr><td>4</td><td>0.10</td><td>0.01</td><td>0.03</td><td>351</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>74971</td></tr><tr><td>macro avg</td><td>0.49</td><td>0.41</td><td>0.41</td><td>74971</td></tr><tr><td>weighted avg</td><td>0.81</td><td>0.86</td><td>0.83</td><td>74971</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.28	0.07	0.12	7766	1	0.83	0.94	0.88	39471	2	0.26	0.03	0.06	552	3	0.95	0.99	0.97	26831	4	0.10	0.01	0.03	351	accuracy			0.86	74971	macro avg	0.49	0.41	0.41	74971	weighted avg	0.81	0.86	0.83	74971	85.96 %	82.53%
	precision	recall	f1-score	support																																												
0	0.28	0.07	0.12	7766																																												
1	0.83	0.94	0.88	39471																																												
2	0.26	0.03	0.06	552																																												
3	0.95	0.99	0.97	26831																																												
4	0.10	0.01	0.03	351																																												
accuracy			0.86	74971																																												
macro avg	0.49	0.41	0.41	74971																																												
weighted avg	0.81	0.86	0.83	74971																																												