

## Model Development Phase Template

Date	03 October 2024
Team ID	LTVIP2024TMID24922
Project Title	Rainfall Prediction 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:

```
logreg = LogisticRegression()  
logreg.fit(X_train_res, y_train_res)
```

```
y_pred2 = logreg.predict(X_test)  
print(confusion_matrix(y_test,y_pred2))  
print(accuracy_score(y_test,y_pred2))  
print(classification_report(y_test,y_pred2))
```

```
svc = SVC()  
svc.fit(X_train_res, y_train_res)
```

```
y_pred5 = svc.predict(X_test)  
print(confusion_matrix(y_test,y_pred5))  
print(accuracy_score(y_test,y_pred5))  
print(classification_report(y_test,y_pred5))
```

```
knn = KNeighborsClassifier(n_neighbors=3)  
knn.fit(X_train_res, y_train_res)
```

```
y_pred4 = knn.predict(X_test)
print(confusion_matrix(y_test,y_pred4))
print(accuracy_score(y_test,y_pred4))
print(classification_report(y_test,y_pred4))
```

```
rf=RandomForestClassifier()
rf.fit(X_train_res,y_train_res)
```

```
y_pred1 = rf.predict(X_test)
print(confusion_matrix(y_test,y_pred1))
print(accuracy_score(y_test,y_pred1))
print(classification_report(y_test,y_pred1))
```

### Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	<pre>print(classification_report(y_test,y_pred1))</pre> <pre> precision    recall  f1-score   support  0           0.88      0.89      0.89      1859 1           0.61      0.57      0.59       541  accuracy          0.82      2400 macro avg         0.74      0.73      0.74      2400 weighted avg      0.82      0.82      0.82      2400</pre>	82%	<pre>print(confusion_matrix(y_test,y_pred1))</pre> <pre> [[1663  196]  [ 235  306]]</pre>
Decision Tree	<pre>print('Classification report {}'.format(classification_report(y_test,y_pred_tree)))</pre> <pre> Classification report      precision    recall  f1-score   support  0           0.80      0.99      0.89      1892 1           0.71      0.89      0.80       508  accuracy          0.80      2400 macro avg         0.75      0.54      0.53      2400 weighted avg      0.78      0.88      0.73      2400</pre>	80%	<pre>print(confusion_matrix(y_test,y_pred_tree))</pre> <pre> [[1872   20]  [ 460   48]]</pre>
K Nearest Neighbour	<pre>print(classification_report(y_test,y_pred4))</pre> <pre> precision    recall  f1-score   support  0           0.91      0.77      0.83      22717 1           0.46      0.72      0.56      6375  accuracy          0.76      29092 macro avg         0.68      0.74      0.70      29092 weighted avg      0.81      0.76      0.77      29092</pre>	75%	<pre>print(confusion_matrix(y_test,y_pred4))</pre> <pre> [[17409  5308]  [ 1808  4567]]</pre>

Logistic Regression	<pre>print(classification_report(y_test,y_pred2))</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>0.92</td><td>0.77</td><td>0.84</td><td>22717</td></tr><tr><td>1</td><td>0.48</td><td>0.76</td><td>0.59</td><td>6375</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>29092</td></tr><tr><td>macro avg</td><td>0.70</td><td>0.77</td><td>0.71</td><td>29092</td></tr><tr><td>weighted avg</td><td>0.82</td><td>0.77</td><td>0.78</td><td>29092</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.92	0.77	0.84	22717	1	0.48	0.76	0.59	6375	accuracy			0.77	29092	macro avg	0.70	0.77	0.71	29092	weighted avg	0.82	0.77	0.78	29092	76%	<pre>print(confusion_matrix(y_test,y_pred2))</pre> <pre>[[17439  5278]  [ 1507 4868]]</pre>
	precision	recall	f1-score	support																													
0	0.92	0.77	0.84	22717																													
1	0.48	0.76	0.59	6375																													
accuracy			0.77	29092																													
macro avg	0.70	0.77	0.71	29092																													
weighted avg	0.82	0.77	0.78	29092																													
XGBoost	<pre>print('Classification report {}'.format(classification_report(y_test,y_predict)))</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>0.87</td><td>0.93</td><td>0.90</td><td>1874</td></tr><tr><td>1</td><td>0.68</td><td>0.52</td><td>0.59</td><td>526</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.84</td><td>2400</td></tr><tr><td>macro avg</td><td>0.78</td><td>0.73</td><td>0.75</td><td>2400</td></tr><tr><td>weighted avg</td><td>0.83</td><td>0.84</td><td>0.83</td><td>2400</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.87	0.93	0.90	1874	1	0.68	0.52	0.59	526	accuracy			0.84	2400	macro avg	0.78	0.73	0.75	2400	weighted avg	0.83	0.84	0.83	2400	84%	<pre>print(confusion_matrix(y_test,y_predict))</pre> <pre>[[1745  129]  [ 250  276]]</pre>
	precision	recall	f1-score	support																													
0	0.87	0.93	0.90	1874																													
1	0.68	0.52	0.59	526																													
accuracy			0.84	2400																													
macro avg	0.78	0.73	0.75	2400																													
weighted avg	0.83	0.84	0.83	2400																													
SVC	<pre>print(classification_report(y_test,y_pred5))</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>0.91</td><td>0.77</td><td>0.83</td><td>1878</td></tr><tr><td>1</td><td>0.47</td><td>0.74</td><td>0.57</td><td>522</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.76</td><td>2400</td></tr><tr><td>macro avg</td><td>0.69</td><td>0.75</td><td>0.70</td><td>2400</td></tr><tr><td>weighted avg</td><td>0.82</td><td>0.76</td><td>0.78</td><td>2400</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.91	0.77	0.83	1878	1	0.47	0.74	0.57	522	accuracy			0.76	2400	macro avg	0.69	0.75	0.70	2400	weighted avg	0.82	0.76	0.78	2400	76%	<pre>print(confusion_matrix(y_test,y_pred5))</pre> <pre>[[1443  435]  [ 136  386]]</pre>
	precision	recall	f1-score	support																													
0	0.91	0.77	0.83	1878																													
1	0.47	0.74	0.57	522																													
accuracy			0.76	2400																													
macro avg	0.69	0.75	0.70	2400																													
weighted avg	0.82	0.76	0.78	2400																													
CatBoost	<pre>print('Classification report {}'.format(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>0</td><td>0.87</td><td>0.95</td><td>0.91</td><td>1880</td></tr><tr><td>1</td><td>0.73</td><td>0.49</td><td>0.59</td><td>520</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.85</td><td>2400</td></tr><tr><td>macro avg</td><td>0.80</td><td>0.72</td><td>0.75</td><td>2400</td></tr><tr><td>weighted avg</td><td>0.84</td><td>0.85</td><td>0.84</td><td>2400</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.87	0.95	0.91	1880	1	0.73	0.49	0.59	520	accuracy			0.85	2400	macro avg	0.80	0.72	0.75	2400	weighted avg	0.84	0.85	0.84	2400	85%	<pre>print(confusion_matrix(y_test,y_pred))</pre> <pre>[[1786  94]  [ 265  255]]</pre>
	precision	recall	f1-score	support																													
0	0.87	0.95	0.91	1880																													
1	0.73	0.49	0.59	520																													
accuracy			0.85	2400																													
macro avg	0.80	0.72	0.75	2400																													
weighted avg	0.84	0.85	0.84	2400																													