

Model Development Phase Template

Date	11-07-2024
Team ID	740047
Project Title	SMOKE DETECTION USING IOT DATASET
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:

Model Building

```
[ ] from sklearn.tree import DecisionTreeClassifier
    from sklearn.metrics import accuracy_score, classification_report
    clf = DecisionTreeClassifier()
```

```
▶ # Train the classifier on the training data
  clf.fit(x_train, y_train)

  # Make predictions on the testing data
  y_pred = clf.predict(x_test)

  # Evaluate the classifier
  report = classification_report(y_test, y_pred)
  print("Classification Report:\n", report)
```

```
🔗 Classification Report:
      precision    recall  f1-score   support

     0       1.00      1.00      1.00       3605
     1       1.00      1.00      1.00       8921

 accuracy          1.00
 macro avg          1.00
 weighted avg       1.00
```

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score, classification_report
```

```
▶ #Initializing the model
model_lr = LogisticRegression()
model_lr.fit(x_train, y_train)
lr_pred_test = model_lr.predict(x_test)
lr_pred_train = model_lr.predict(x_train)
test_acc_lr = accuracy_score(y_test, lr_pred_test)
train_acc_lr = accuracy_score(y_train, lr_pred_train)
print('Logistic Regression test accuracy: ', test_acc_lr)
print(classification_report(y_test, lr_pred_test))
```

```
⇒ Logistic Regression test accuracy: 0.9691840970780776
      precision    recall  f1-score   support

      0         0.94      0.95      0.95         3605
      1         0.98      0.98      0.98         8921

   accuracy                   0.97         12526
  macro avg              0.96      0.96      0.96         12526
 weighted avg              0.97      0.97      0.97         12526
```

KNN classifier

✓
4s

```
▶ from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
knn_pred_test = knn.predict(x_test)
knn_pred_train = knn.predict(x_train)
test_acc_knn = accuracy_score(y_test, knn_pred_test)
train_acc_knn = accuracy_score(y_train, knn_pred_train)
print('KNN Test Accuracy is: ', test_acc_knn)
print(classification_report(y_test, knn_pred_test))
```

```
⇒ KNN Test Accuracy is: 0.9992814944914578
      precision    recall  f1-score   support

      0         1.00      1.00      1.00         3605
      1         1.00      1.00      1.00         8921

   accuracy                   1.00         12526
  macro avg              1.00      1.00      1.00         12526
 weighted avg              1.00      1.00      1.00         12526
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decision tree classifier	<pre> Classification Report: precision recall f1-score support 0 1.00 1.00 1.00 3605 1 1.00 1.00 1.00 8921 accuracy 1.00 1.00 1.00 12526 macro avg 1.00 1.00 1.00 12526 weighted avg 1.00 1.00 1.00 12526 </pre>	0.99	<pre> confusion_matrix(y_test,y_pred) array([[3605, 0], [1, 8920]]) </pre>
Logistic Regression	<pre> Logistic Regression test accuracy: 0.9691848978780776 precision recall f1-score support 0 0.94 0.95 0.95 3605 1 0.98 0.98 0.98 8921 accuracy 0.96 0.96 0.97 12526 macro avg 0.96 0.96 0.96 12526 weighted avg 0.97 0.97 0.97 12526 </pre>	0.96	<pre> confusion_matrix(y_test,lr_pred_test) array([[3437, 168], [218, 8703]]) </pre>
K-Nearest Neighbors	<pre> KNN Test Accuracy is: 0.9992814944914578 precision recall f1-score support 0 1.00 1.00 1.00 3605 1 1.00 1.00 1.00 8921 accuracy 1.00 1.00 1.00 12526 macro avg 1.00 1.00 1.00 12526 weighted avg 1.00 1.00 1.00 12526 </pre>	0.99	<pre> confusion_matrix(y_test,knn_pred_test) array([[3599, 6], [3, 8918]]) </pre>