Performance & Final Submission Phase

Model Performance Test

Date	19 November 2023
Team ID	Team - 593188
Project Name	STARTUP PROPHET: HARNESSING AI TO DIVINE THE FUTURE OF STARTUP SUCCESS
Maximum Marks	10 marks

Model Performance Testing:

Regression Model-

Activity 1: Logistic Regression

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

    # Create a Logistic Regression model
    logistic_model = LogisticRegression()
    logistic_model.fit(X_train, y_train)

    # Predict on the test set
    y_pred = logistic_model.predict(X_test)

    # Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    conf_matrix = confusion_matrix(y_test, y_pred)
    class_report = classification_report(y_test, y_pred)

    print("Accuracy:", accuracy)
    print("\nConfusion Matrix:\n", conf_matrix)
    print("\nClassification Report:\n", class_report)
```

```
Accuracy: 0.6389891696750902

Confusion Matrix:
 [[  0 100]
 [  0 177]]

Classification Report:
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00       100
         1.0       0.64      1.00      0.78       177

    accuracy                           0.64       277
   macro avg       0.32      0.50      0.39       277
weighted avg       0.41      0.64      0.50       277
```

Activity 2: Support vector machine

```
[ ] from sklearn.svm import SVC
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

    # Create an SVM classifier with RBF kernel
    svm_classifier = SVC(kernel='rbf', random_state=42)

    # Train the SVM classifier on the balanced dataset
    svm_classifier.fit(X_train, y_train)

    # Predict on the test set
    y_pred_svm = svm_classifier.predict(X_test)

    # Evaluate the SVM classifier
    accuracy_svm = accuracy_score(y_test, y_pred_svm)
    conf_matrix_svm = confusion_matrix(y_test, y_pred_svm)
    class_report_svm = classification_report(y_test, y_pred_svm)

    # Print the evaluation metrics
    print("SVM Accuracy:", accuracy_svm)
    print("\nSVM Confusion Matrix:\n", conf_matrix_svm)
    print("\nSVM Classification Report:\n", class_report_svm)
```

```
SVM Accuracy: 0.6389891696750902

SVM Confusion Matrix:
 [[  0 100]
 [  0 177]]

SVM Classification Report:
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00       100
         1.0       0.64      1.00      0.78       177

    accuracy                           0.64       277
   macro avg       0.32      0.50      0.39       277
weighted avg       0.41      0.64      0.50       277
```

Classification Model-

Activity 2: Random forest model

```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

    # Create a Random Forest classifier
    RF = RandomForestClassifier(random_state=42)

    # Train the Random Forest classifier on the balanced dataset
    RF.fit(X_train, y_train)

    # Predict on the test set
    y_pred_rf = RF.predict(X_test)

    # Evaluate the Random Forest classifier
    accuracy_rf = accuracy_score(y_test, y_pred_rf)
    conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)
    class_report_rf = classification_report(y_test, y_pred_rf)

    # Print the evaluation metrics
    print("Random Forest Accuracy:", accuracy_rf)
    print("\nRandom Forest Confusion Matrix:\n", conf_matrix_rf)
    print("\nRandom Forest Classification Report:\n", class_report_rf)
```

```
Random Forest Accuracy: 1.0

Random Forest Confusion Matrix:
 [[100   0]
 [  0 177]]

Random Forest Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00       100
         1.0       1.00      1.00      1.00       177

    accuracy                           1.00       277
   macro avg       1.00      1.00      1.00       277
weighted avg       1.00      1.00      1.00       277
```