Rock vs Mine Prediction - Detailed Documentation

PROJECT OVERVIEW

This project builds a Machine Learning model to classify sonar signals as either ROCK or MINE.

Using Logistic Regression, the model learns from signal patterns to differentiate between the two.

DATASET DETAILS

- Dataset: sonar_data.csv
- Rows: 208 samples
- Columns: 61 (60 features + 1 label)
- Labels: R (Rock), M (Mine)

Each row represents sonar readings bounced off objects. The last column indicates whether the object is a rock or a mine.

PROJECT FLOW

----

1. IMPORT LIBRARIES

   Import numpy, pandas, sklearn.model_selection, sklearn.linear_model, sklearn.metrics.

2. LOAD DATA

   Load sonar_data.csv using pandas.

3. EXPLORE DATA

   - Check data shape
   - Generate statistical summary using describe()

- Count Rock and Mine samples using value_counts()

4. SEPARATE FEATURES AND LABELS

   - X = sonar_data.drop(columns=60, axis=1)
   - y = sonar_data[60]

5. SPLIT DATA

   - train_test_split(X, y, test_size=0.1, stratify=y, random_state=1)
   - Ensures 90% training and 10% testing split.

6. MODEL CREATION AND TRAINING

   - model = LogisticRegression()
   - model.fit(X_train, y_train)

7. MODEL EVALUATION

   - Predict and calculate accuracy on training and test data
   - accuracy_score(y_true, y_pred)

CODE EXPLANATION

```
model = LogisticRegression()
model.fit(x_train, y_train)
```

- Creates a logistic regression model
- Trains it using training data

```
x_train_prediction = model.predict(x_train)
training_accuracy = accuracy_score(y_train, x_train_prediction)
```

- Tests model performance on training data

```
x_test_prediction = model.predict(x_test)
test_accuracy = accuracy_score(y_test, x_test_prediction)
```

- Evaluates model performance on unseen test data

RESULTS SUMMARY

Training Accuracy: 83%

Testing Accuracy: 76%

The model performs well with a small drop between train/test accuracy (good generalization).

FLOW OF EXECUTION (Simplified)

----

1 Import libraries

2 Load dataset

3 Explore data

4 Split into features and labels

5 Train/test split

6 Create Logistic Regression model

7 Train model

8 Evaluate accuracy

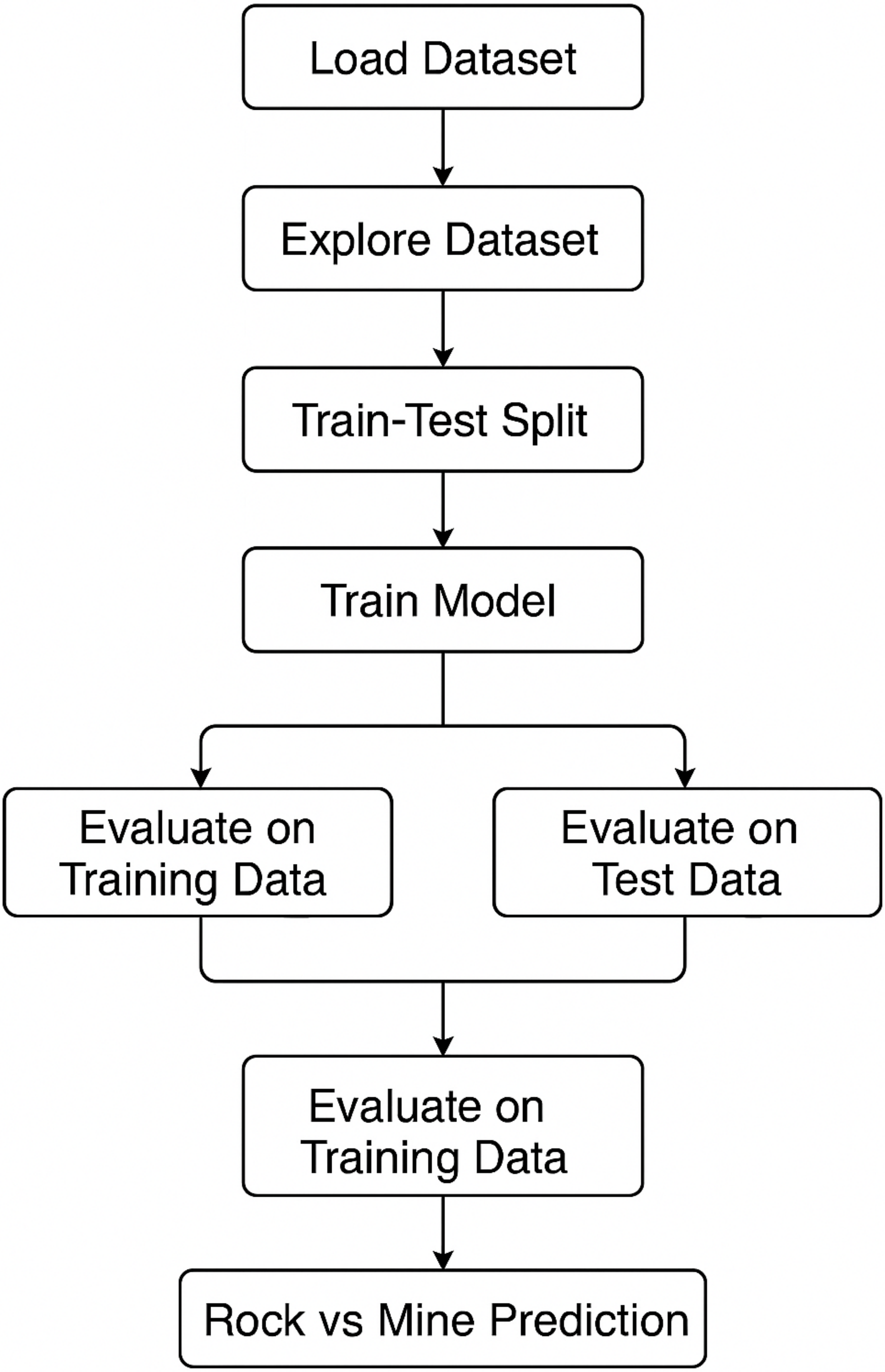
9 Interpret results

SUGGESTED IMPROVEMENTS

- Apply feature scaling using StandardScaler.
- Tune hyperparameters (C, solver).
- Compare performance with other models (SVM, RandomForest, KNN).
- Use cross-validation for robustness.
- Visualize confusion matrix and ROC curves.

CONCLUSION

The Rock vs Mine model demonstrates the complete ML workflow   data processing, model training, and evaluation. Logistic Regression provides a solid baseline for this classification problem.


Load Dataset
Explore Dataset
Train-Test Split
Train Model
Evaluate on Training Data
Evaluate on Test Data
Evaluate on Training Data
Rock vs Mine Prediction