SUPERIOR UNIVERSITY

NAME: MINAHIL IQBAL

ROLL NO: 062

SUBJECT: PAI (LAB)

SECTION: BSAI-4B

SUBMITTED TO: SIR RASIKH

Machine Learning Model Report: Spaceship Titanic Dataset

Introduction

This report analyzes the provided Python code, which applies machine learning techniques to the Spaceship Titanic dataset. The code performs data preprocessing, model training, and prediction using Random Forest Classifier and SVC (Support Vector Classifier).

Code Breakdown

1. Data Loading

- The training dataset (train.csv) and test dataset (test.csv) are loaded using pandas.read_csv ().
- train_data.info () and test_data.info() are used to inspect the dataset structure.

2. Feature Selection

- The target variable is Transported.
- Selected features for training: PassengerId, Transported.

3. Data Preprocessing

- Label Encoding is applied to the Transported column using LabelEncoder.
- The Transported column from the training set is copied into the test dataset (which is not standard practice in real-world scenarios).
- Features are extracted, and a new CSV file (test88.csv) is created from the processed test dataset.

4. Model Training

- The Support Vector Classifier (SVC) model (model_svc) is used for training.
- However, RandomForestClassifier is imported but never used.
- Model training lacks data splitting (e.g., train-test split), which is essential for model evaluation.

5. Predictions and Submission File Creation

- Predictions are made using model_svc.predict (test10_data).
- A submission file (submission0.csv) is created, containing PassengerId and predicted Transported values.

Observations & Recommendations

Strengths:

- Proper use of Pandas for data handling.
- Implementation of Label Encoding for categorical variables.
- Use of Scikit-Learn's SVC model for classification.
- Proper dataset loading and extraction of relevant features.

Missing Train-Test Split:

A train-validation split using train_test_split() is necessary to evaluate the model before applying it to test data.

Model Selection:

- SVC might not be the best choice for classification in this dataset.
- RandomForestClassifier (which was imported but not used) would likely provide better results.

Model Performance Evaluation:

- The code does not measure accuracy or any other performance metrics
- Adding an accuracy score (accuracy_score(y_test, predictions)) or confusion matrix (confusion_matrix(y_test, predictions)) would improve evaluation.

Conclusion

The code provides a basic pipeline for training a machine learning model, but it contains major issues in feature selection, test data handling, and model evaluation. Improving these aspects and using an appropriate classification model will enhance the overall predictive performance.

OUTPUT

Delimiter: ,

	PassengerId	Transported
1	0013_01	0
2	0018_01	1
3	0019_01	0
4	0021_01	0
5	0023_01	1
6	0027_01	1
7	0029_01	1
8	0032_01	1
9	0032_02	1
10	0033_01	1
11	0037_01	1
12	0040_01	1
13	0040_02	1
14	0042_01	0
15	0046_01	1
16	0046_02	0
17	0046_03	0
18	0047_01	0
19	0047_02	1
20	0047_03	1

Delimiter: ,

	PassengerId	Transported
1	0013_01	False
2	0018_01	False
3	0019_01	False
4	0021_01	False
5	0023_01	False
6	0027_01	False
7	0029_01	False
8	0032_01	False
9	0032_02	False
10	0033_01	False
11	0037_01	False
12	0040_01	False
13	0040_02	False
14	0042_01	False
15	0046_01	False
16	0046_02	False
17	0046_03	False
18	0047_01	False
19	0047_02	False
20	0047_03	False