

pythonProject1

Version control

Current File

MachineFailurePredictorPipeline.py

AlexNetPredictorPipeline.py

Run MachineFailurePredictorPipeline

Printing Train Results

ML Trained Model	Best Set of Parameter Values	MCC-score on the 5-fold
Cross Validation on Training Data (80%)		
Multi-layer Neural Network	{'activation': 'relu', 'hidden_layer_sizes': (100,),'learning_rate': 'adaptive'}	0.802920093328211
Support Vector Machine	{'C': 10, 'gamma': 0.1, 'kernel': 'poly'}	0.7983294268474874
K-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 3, 'p': 1}	0.7397595078168042
Decision Tree	{'ccp_alpha': 0.0, 'criterion': 'entropy', 'max_depth': 5}	0.8071282624480309
Softmax Regression	{'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}	0.6324552683237885

Step 8A: Test model
Step 8B: Test model
Step 8C: Test model
Step 8D: Test model

pythonProject1 > MachineFailurePredictorPipeline.py 1:1 LF UTF-8 4 spaces Python 3.12 (pythonProject1)

pythonProject1

Version control

Current File

MachineFailurePredictorPipeline.py

AlexNetPredictorPipeline.py

Run MachineFailurePredictorPipeline

test('machine_failure') = y_test
test.to_csv(test_file_name, index=False)

Cross Validation on Testing Data (20%)

ML Trained Model	Best Set of Parameter Values	MCC-score on the 5-fold
Multi-layer Neural Network	{'activation': 'relu', 'hidden_layer_sizes': (100,),'learning_rate': 'constant'}	0.7050707456243541
Support Vector Machine	{'C': 10, 'gamma': 0.1, 'kernel': 'poly'}	0.8376536252077308
K-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 3, 'p': 1}	0.7146122136752029
Decision Tree	{'ccp_alpha': 0.0, 'criterion': 'entropy', 'max_depth': 5}	0.8207902411555048
Softmax Regression	{'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}	0.6129810409717638

The model with the highest MCC score is Support Vector Machine with a score of 0.8376536252077308. This is the model that should be used in the future for machine failure prediction.

Process finished with exit code 0

pythonProject1 > MachineFailurePredictorPipeline.py 72:1 LF UTF-8 4 spaces Python 3.12 (pythonProject1)

Printing Train Results		
ML Trained Model	Best Set of Parameter Values	MCC-score on the 5-fold Cross Validation on Training Data (80%)
Multi-layer Neural Network	{'activation': 'relu', 'hidden_layer_sizes': (100,),'learning_rate': 'adaptive'}	0.7836525735602641
Support Vector Machine	{'C': 10, 'gamma': 0.1, 'kernel': 'poly'}	0.7983294268474874
K-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 3, 'p': 1}	0.7397595078168042
Decision Tree	{'ccp_alpha': 0.0, 'criterion': 'entropy', 'max_depth': 5}	0.810609899189139
Softmax Regression	{'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}	0.6324552683237885
Step 8A: Test model Step 8B: Test model Step 8C: Test model Step 8D: Test model Step 8E: Test model		
ML Trained Model	Best Set of Parameter Values	MCC-score on the 5-fold Cross Validation on Testing Data (20%)
Multi-layer Neural Network	{'activation': 'relu', 'hidden_layer_sizes': (100,),'learning_rate': 'adaptive'}	0.8128254011095852
Support Vector Machine	{'C': 10, 'gamma': 0.1, 'kernel': 'poly'}	0.8376536252077308
K-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 3, 'p': 1}	0.7146122136752029
Decision Tree	{'ccp_alpha': 0.0, 'criterion': 'entropy', 'max_depth': 5}	0.8357698854145772
Softmax Regression	{'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}	0.6129810409717638