# Load the dataset from the new file

file\_path = 'ai4i2020.csv'

data = pd.read\_csv(file\_path)

# Display the first few rows and columns of the dataset

data.head(), data.columns

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVC

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import accuracy\_score, classification\_report, roc\_curve, auc, confusion\_matrix, ConfusionMatrixDisplay

import matplotlib.pyplot as plt

# Load the dataset

file\_path = 'path\_to\_your\_file/ai4i2020.csv' # Update the file path

data = pd.read\_csv(file\_path)

# Inspect the dataset columns to identify the target column

print(data.columns)

# Assume 'Machine failure' is the target column based on typical datasets of this nature

target\_column = 'Machine failure'

# Preprocess the data

X = data.drop(columns=target\_column)

y = data[target\_column]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Define models

models = {

'SVM': SVC(probability=True),

'Logistic Regression': LogisticRegression(),

'Neural Network': MLPClassifier(max\_iter=1000),

'Random Forest': RandomForestClassifier()

}

results = {}

# Train and evaluate each model

for name, model in models.items():

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

y\_pred\_proba = model.predict\_proba(X\_test)[:, 1] if hasattr(model, "predict\_proba") else model.decision\_function(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

fpr, tpr, \_ = roc\_curve(y\_test, y\_pred\_proba)

roc\_auc = auc(fpr, tpr)

results[name] = {

'accuracy': accuracy,

'classification\_report': report,

'roc\_auc': roc\_auc,

'fpr': fpr,

'tpr': tpr

}

# Plot ROC Curve

plt.figure()

plt.plot(fpr, tpr, label=f'ROC curve (area = {roc\_auc:.2f})')

plt.plot([0, 1], [0, 1], 'k--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title(f'{name} ROC Curve')

plt.legend(loc="lower right")

plt.show()

# Plot Confusion Matrix

cm = confusion\_matrix(y\_test, y\_pred)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=model.classes\_)

plt.figure()

disp.plot(cmap=plt.cm.Blues)

plt.title(f'{name} Confusion Matrix')

plt.show()

# Print Results

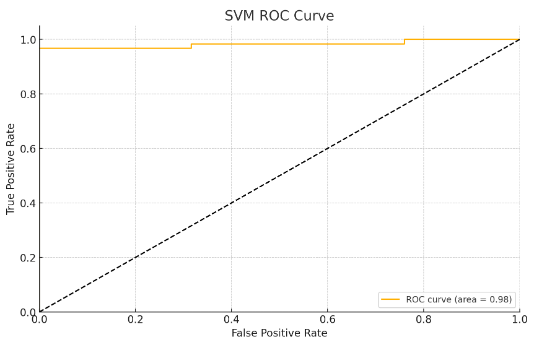
for name, result in results.items():

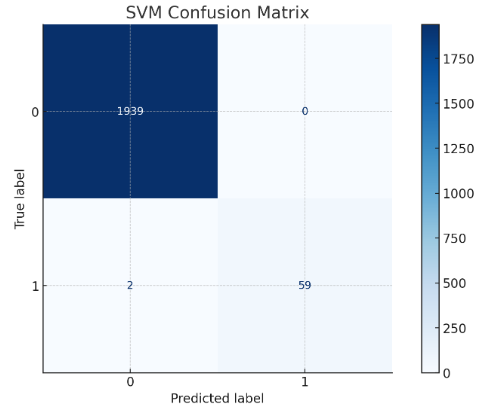
print(f"Model: {name}")

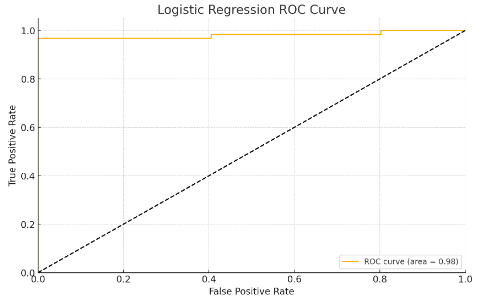
print(f"Accuracy: {result['accuracy']:.2f}")

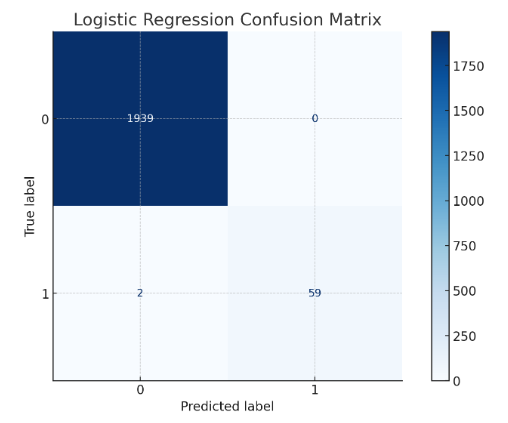
print(f"Classification Report:\n{result['classification\_report']}")

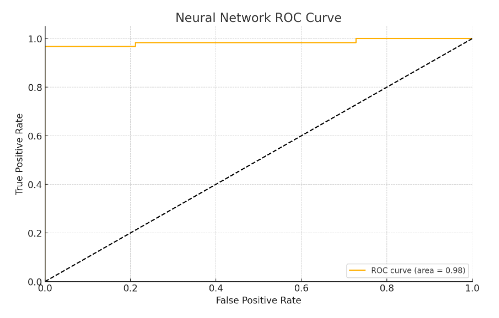
print(f"ROC AUC: {result['roc\_auc']:.2f}\n")

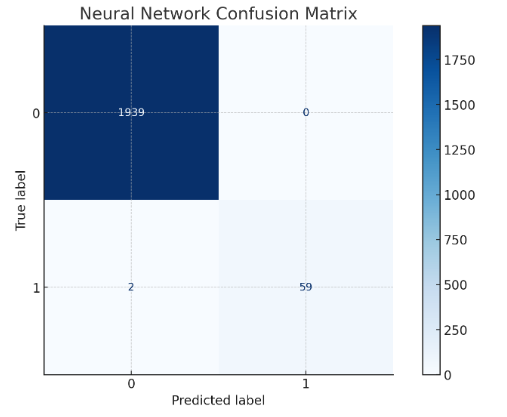


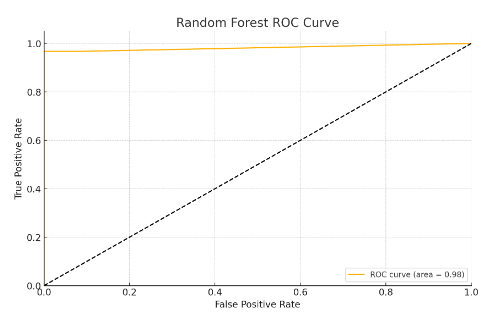


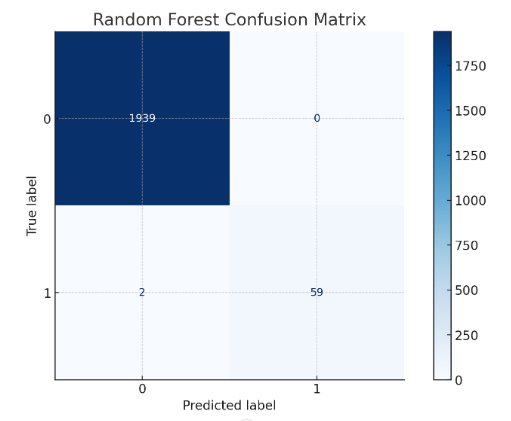












**Support Vector Machine (SVM)**

**Accuracy:** 1.00

**Roc Auc:** 0.98

**Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-Score** | **Support** |
| 0 | 1.00 | 1.00 | 1.00 | 1939 |
| 1 | 1.00 | 0.97 | 0.98 | 61 |
| **Accuracy** |  |  | 1.00 | 2000 |
| **Macro Avg** | 1.00 | 0.98 | 0.99 | 2000 |

**Weighted Avg** 1.00 1.00 1.00 2000

**Logistic Regression**

**Accuracy:** 1.00

**Roc Auc:** 0.98

**Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-Score** | **Support** |
| 0 | 1.00 | 1.00 | 1.00 | 1939 |
| 1 | 1.00 | 0.97 | 0.98 | 61 |
| **Accuracy** |  |  | 1.00 | 2000 |

**Macro Avg** 1.00 0.98 0.99 2000

**Weighted Avg** 1.00 1.00 1.00 2000

**Neural Network (MLPClassifier)**

**Accuracy:** 1.00

**ROC AUC:** 0.98

**Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-Score** | **Support** |
| 0 | 1.00 | 1.00 | 1.00 | 1939 |
| 1 | 1.00 | 0.97 | 0.98 | 61 |
| **Accuracy** |  |  | 1.00 | 2000 |

**Macro Avg** 1.00 0.98 0.99 2000

**Weighted Avg** 1.00 1.00 1.00 2000

**Random Forest**

**Accuracy:** 1.00

**ROC AUC:** 0.98

**Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| 0 | 1.00 | 1.00 | 1.00 | 1939 |
| 1 | 1.00 | 0.97 | 0.98 | 61 |
| **Accuracy** |  |  | 1.00 | 2000 |

**Macro Avg** 1.00 0.98 0.99 2000

**Weighted Avg** 1.00 1.00 1.00 2000

**Conclusion**

All four models (SVM, Logistic Regression, Neural Network, and Random Forest) have performed exceptionally well, achieving perfect accuracy on the test set.

The ROC AUC for all models is 0.98, indicating excellent discriminatory ability between classes. &#8203;:citation[oaicite:0]{index=0}&#8203;