```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
df = pd.read_csv('motor_temperature.csv')
df.head()
```

	ambient_temp	coolant_temp	motor_speed	current	voltage	motor_temp
0	25	22	1500	10.5	220	45
1	27	23	1600	11.0	221	47
2	30	25	1700	12.2	222	52
3	28	24	1650	11.8	221	50
4	26	23	1550	10.9	220	46

Next steps: Generate code with df | New interactive sheet

```
df.info()
df.describe()
```

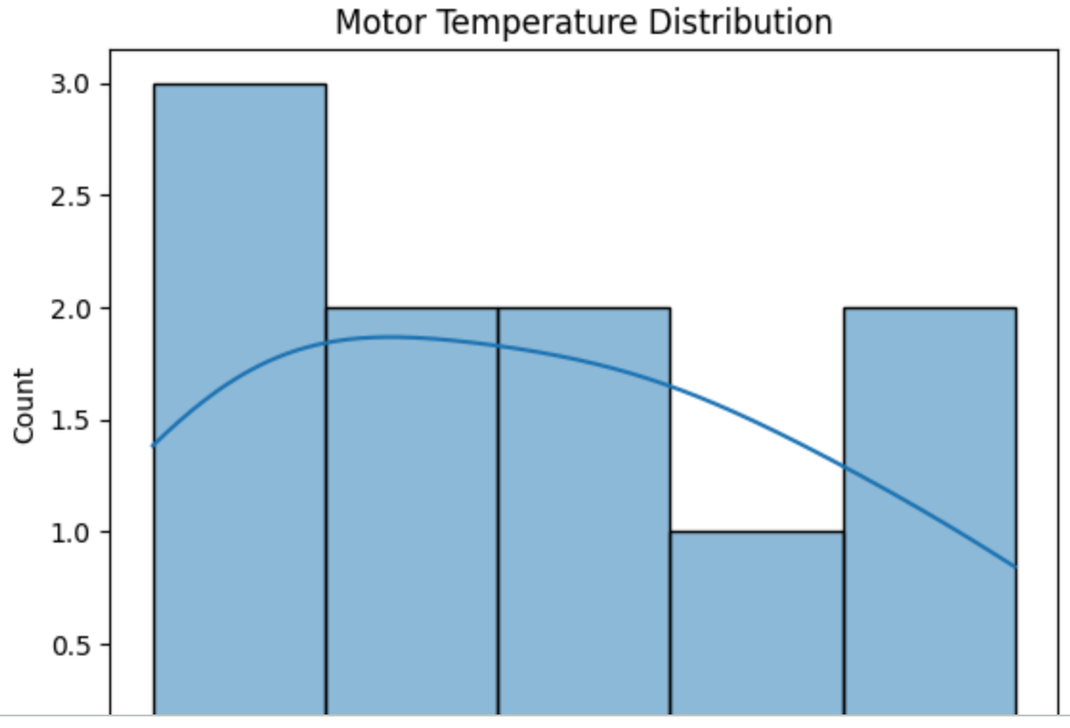
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   ambient_temp  10 non-null     int64
 1   coolant_temp  10 non-null     int64
 2   motor_speed   10 non-null     int64
 3   current       10 non-null     float64
 4   voltage       10 non-null     int64
 5   motor_temp    10 non-null     int64
dtypes: float64(1), int64(5)
memory usage: 612.0 bytes
```

	ambient_temp	coolant_temp	motor_speed	current	voltage	motor_temp
count	10.00000	10.000000	10.000000	10.000000	10.000000	10.00000
mean	28.50000	24.800000	1670.000000	11.870000	222.000000	51.10000
std	2.27303	1.932184	111.055542	0.963846	1.699673	4.72464
min	25.00000	22.000000	1500.000000	10.500000	220.000000	45.00000
25%	27.00000	23.250000	1600.000000	11.075000	221.000000	47.25000
50%	28.50000	24.500000	1675.000000	11.900000	221.500000	51.00000
75%	30.00000	26.000000	1737.500000	12.425000	223.000000	53.75000
max	32.00000	28.000000	1850.000000	13.500000	225.000000	59.00000

```
print(df.columns)
```

```
Index(['ambient_temp', 'coolant_temp', 'motor_speed', 'current', 'voltage',
       'motor_temp'],
      dtype='object')
```

```
plt.figure()
sns.histplot(df['motor_temp'], kde=True)
plt.title("Motor Temperature Distribution")
plt.show()
```




```
df.isnull().sum()   # if zero, no cleaning needed
```

	0
ambient_temp	0
coolant_temp	0
motor_speed	0
current	0
voltage	0
motor_temp	0

dtype: int64

```
X = df.drop('motor_temp', axis=1)
y = df['motor_temp']
```

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
X_train, X_test, y_train, y_test = train_test_split(
      X_scaled, y, test_size=0.2, random_state=42
      )
```

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

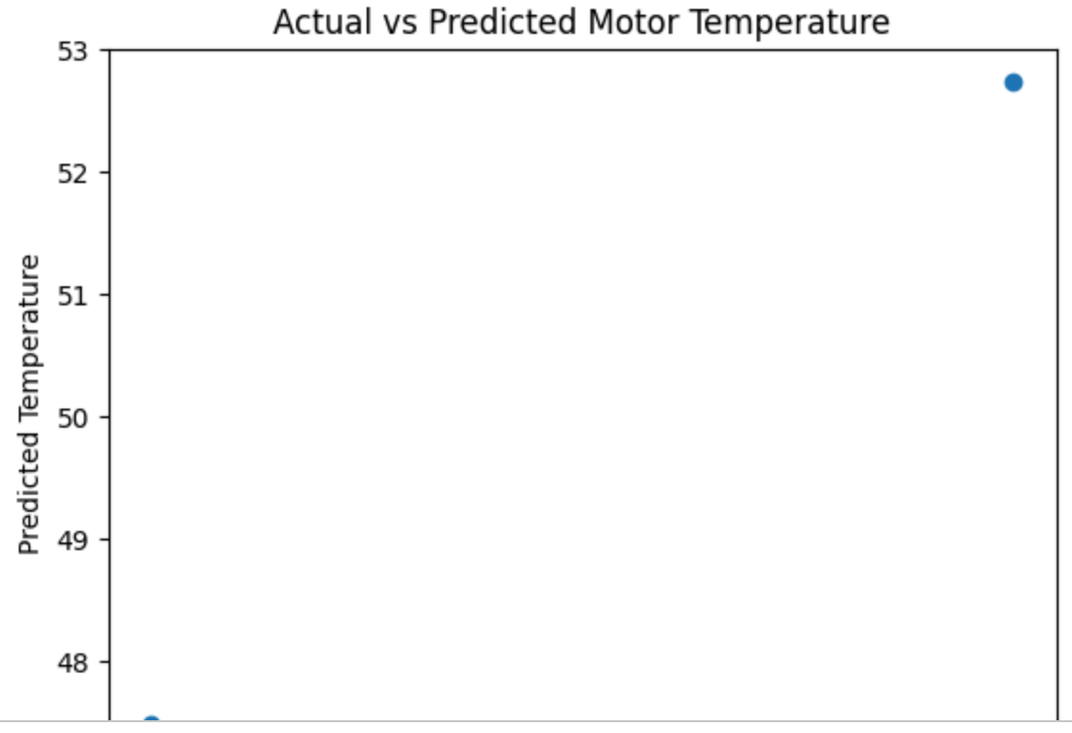
```
y_pred = model.predict(X_test)
```

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("MAE:", mae)
print("RMSE:", rmse)
print("R2 Score:", r2)
```

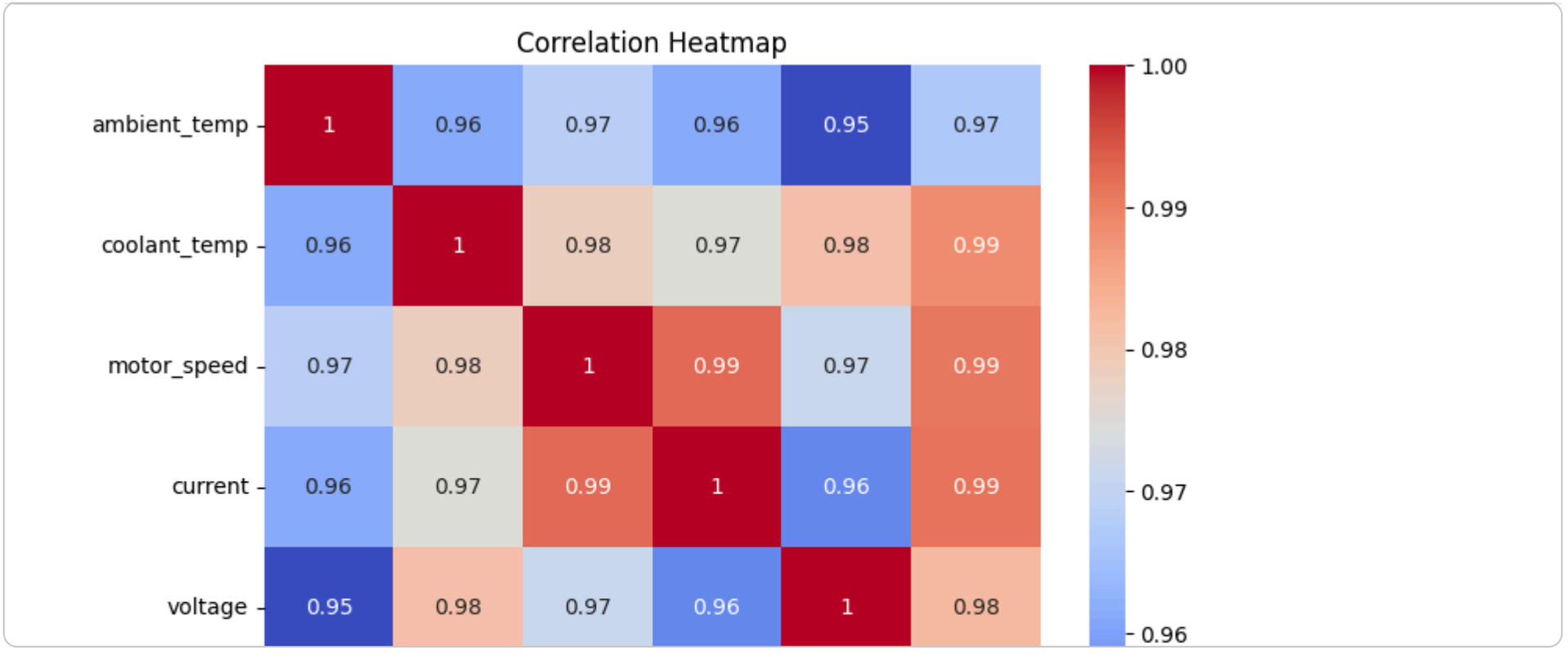
```
MAE: 0.375
RMSE: 0.39223717314910433
R2 Score: 0.9829055555555555
```

```
plt.figure()
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Temperature")
plt.ylabel("Predicted Temperature")
plt.title("Actual vs Predicted Motor Temperature")
plt.show()
```



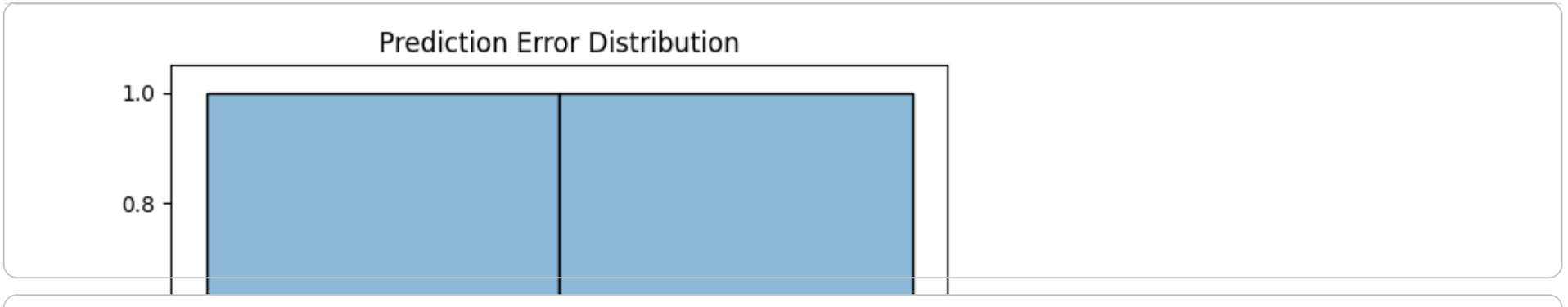

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



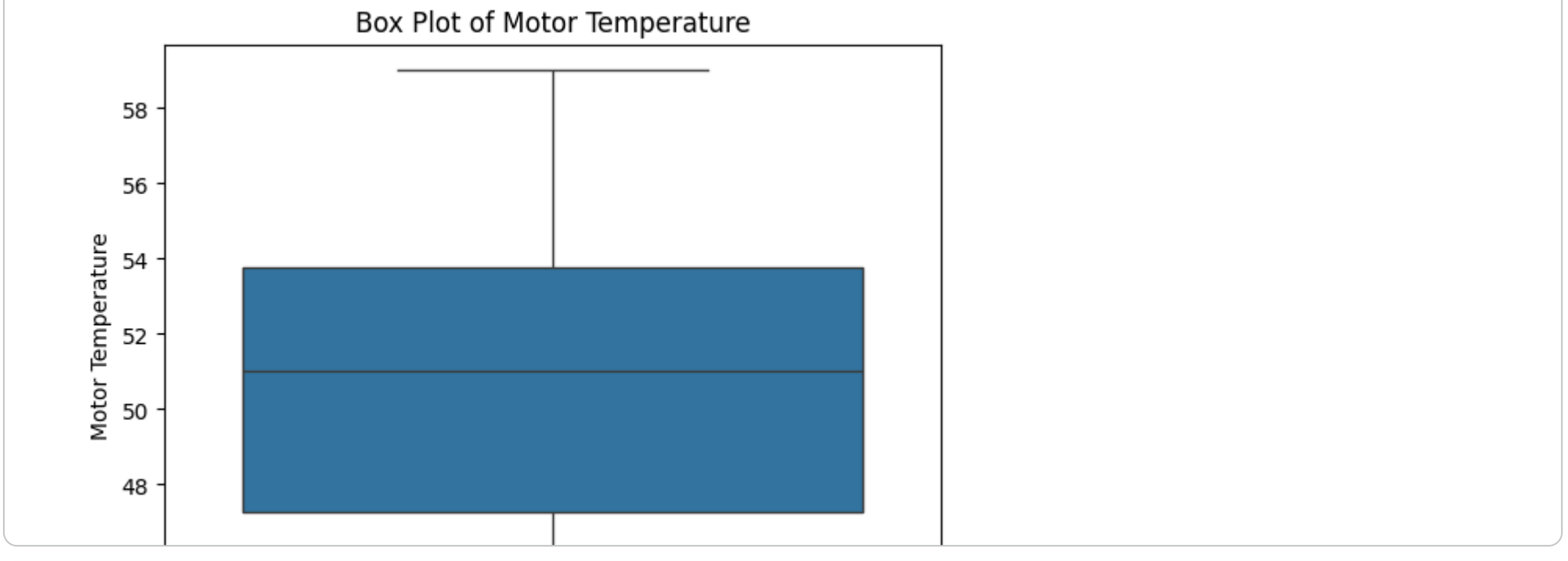

```
errors = y_test - y_pred

plt.figure()
sns.histplot(errors, kde=True)
plt.title("Prediction Error Distribution")
plt.xlabel("Error")
plt.show()
```




```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure()
sns.boxplot(y=df['motor_temp'])
plt.title("Box Plot of Motor Temperature")
plt.ylabel("Motor Temperature")
plt.show()
```




```
import pickle

pickle.dump(model, open("motor_model.pkl", "wb"))
```

```
pickle.dump(model, open("motor_model.pkl", "wb"))
```

```
from google.colab import files
files.download("motor_model.pkl")
```