***Title : Predictive maintenance for machinery***

***Program***

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, GridSearchCV

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import (

classification\_report,

accuracy\_score,

confusion\_matrix,

roc\_auc\_score,

roc\_curve

)

n\_samples = 200

data={

'temperature’: np.random.normal(75, 5, n\_samples),

'vibration': np.random.normal(0.03, 0.01, n\_samples),

'pressure': np.random.normal(100, 3, n\_samples),

'failure': np.random.choice([0, 1], size=n\_samples, p=[0.8, 0.2])

}

df=pd.DataFrame(data)

X=df[['temperature', 'vibration', 'pressure']]

y=df['failure']

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=RandomForestClassifier(n\_estimators=100, random\_state=42)

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

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

y\_pred=model.predict\_proba(X\_test)[:, 1]

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

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

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

feature\_importance=model.feature\_importances\_

sns.barplot(x=feature\_importance, y=['temperature', 'vibration', 'pressure'])

plt.title("Feature Importances")

plt.show()

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

plt.plot(fpr, tpr, label=f ”AUC={roc\_auc\_score(y\_test, y\_prob):.2f}")

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

plt.xlabel("Palse Positive Rate")

plt.ylabel("True Positive Rate")

plt.title("ROC Curve")

plt.legend()

plt.show()

sample=pd.DataFrame({

'temperature': [80],

'vibration': [0.045),

'pressure': [105]

})

sample\_scaled=scaler.transform(sample)

prediction model.predict(sample\_scaled)

print("Prediction:", "maintenance Needed" if prediction [0]==1 else "Machine Healthy")