IBM AICTE PROJECT

PREDICTIVE MAINTENANCE OF INDUSTRIAL MACHINERY

Presented By:

Student name : ADHITHYA LP
College Name & Department : JERUSALEM COLLEGE OF ENGINEERING B Tech Computer science and business systems

edunet foundation

OUTLINE

- Problem Statement
- Technology used
- Wow factor
- End users
- Result
- Conclusion
- Git-hub Link
- Future scope
- IBM Certifications

PROBLEM STATEMENT

Industrial machinery can unexpectedly fail due to issues like tool wear, overheating, or power faults. Such failures lead to unplanned downtime and increased maintenance costs. Predicting these failures in advance using machine sensor data is critical for proactive maintenance. The goal is to build a classification model to predict the specific type of failure before it happens.

- Proposed Solution:

  We analysed real-time sensor data (temperature, torque, speed, etc.) from machines. Pre-processing included encoding, scaling, and balancing the dataset using SMOTE. A Random Forest Classifier was trained to predict six types of machine failure. The model achieved over 99% accuracy, supporting reliable predictive maintenance decisions.

TECHNOLOGY USED

- IBM cloud lite services
- Imbalanced-learn (SMOTE)
- Pandas & NumPy
- Matplotlib & seaborn

IBM CLOUD SERVICES USED

- IBM Cloud Watsonx AI Studio
- IBM Cloud Watsonx AI runtime
- IBM Watsonx Machine learning
- IBM Cloud functions

WOW FACTORS

Achieved exceptionally high accuracy (99%) in predicting multiple types of industrial failures using real-time sensor data. Empowers industries to perform proactive maintenance, significantly reducing downtime and saving operational costs.

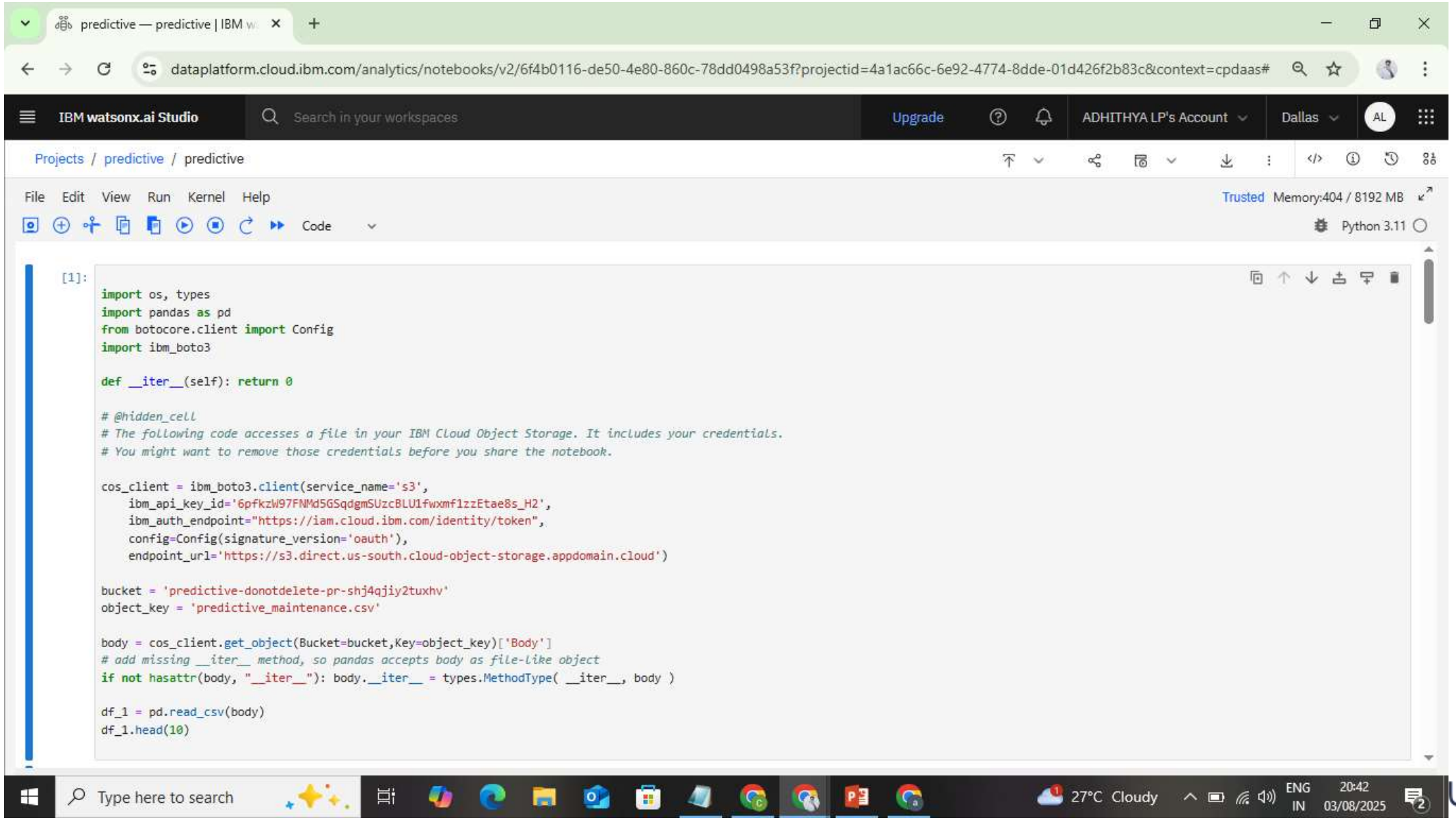
Unique features:

- Predicts specific failure types.
- Uses real-time sensor metrics like temperature, torque and wear.
- Incorporates SMOTE to handle class imbalance for better learning.
- Built with Random Forest Classifier for accuracy and interpretability.
- Scalable to live environments with IBM Cloud deployment options.
- Achieves near perfect precision and recall, making it production-ready.

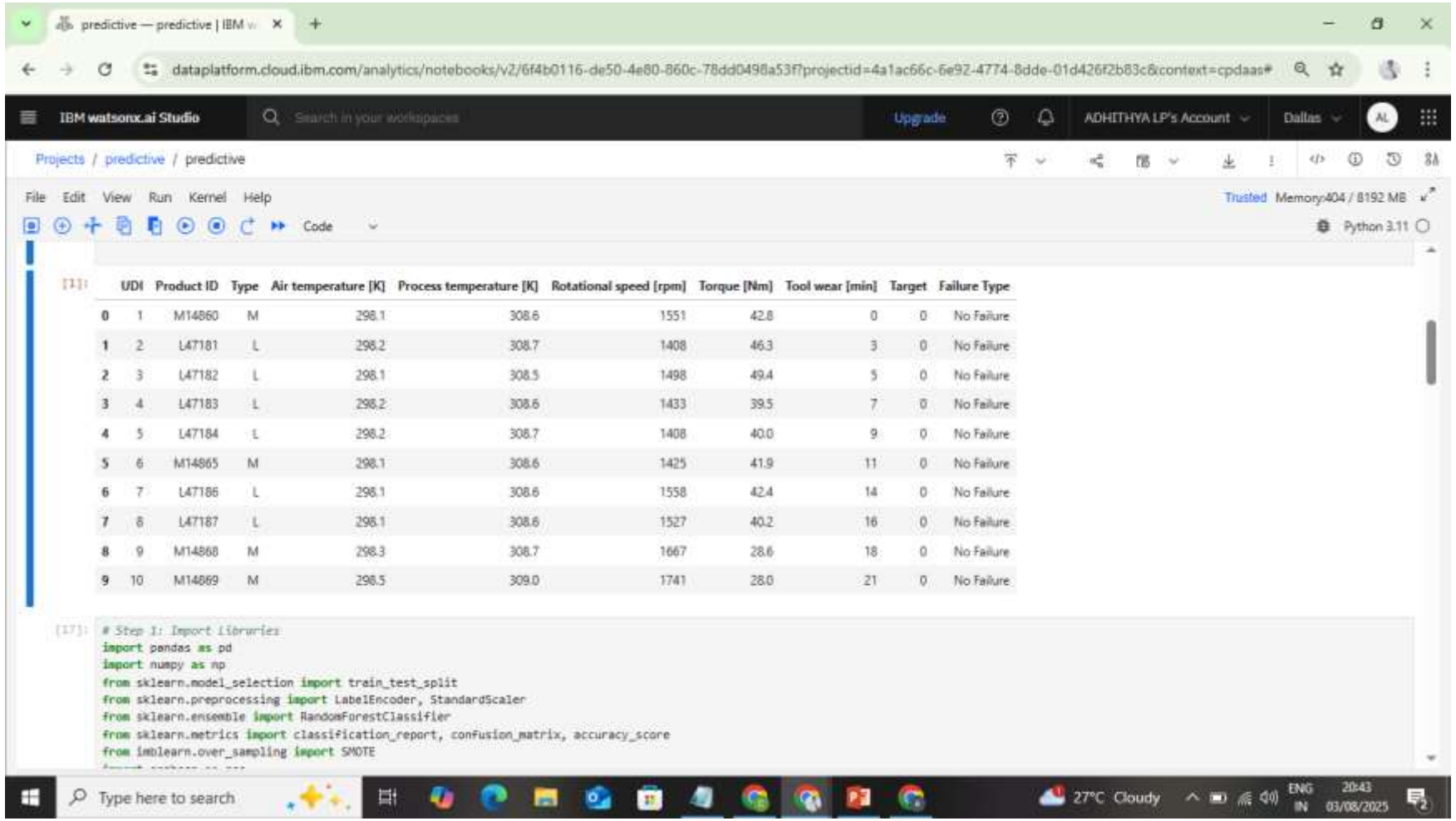
END USERS

- Plant/Factory Managers
- Industrial Automation Companies
- Data Analysts in Manufacturing
- IoT Platform Providers
- Maintenance Engineers

RESULTS

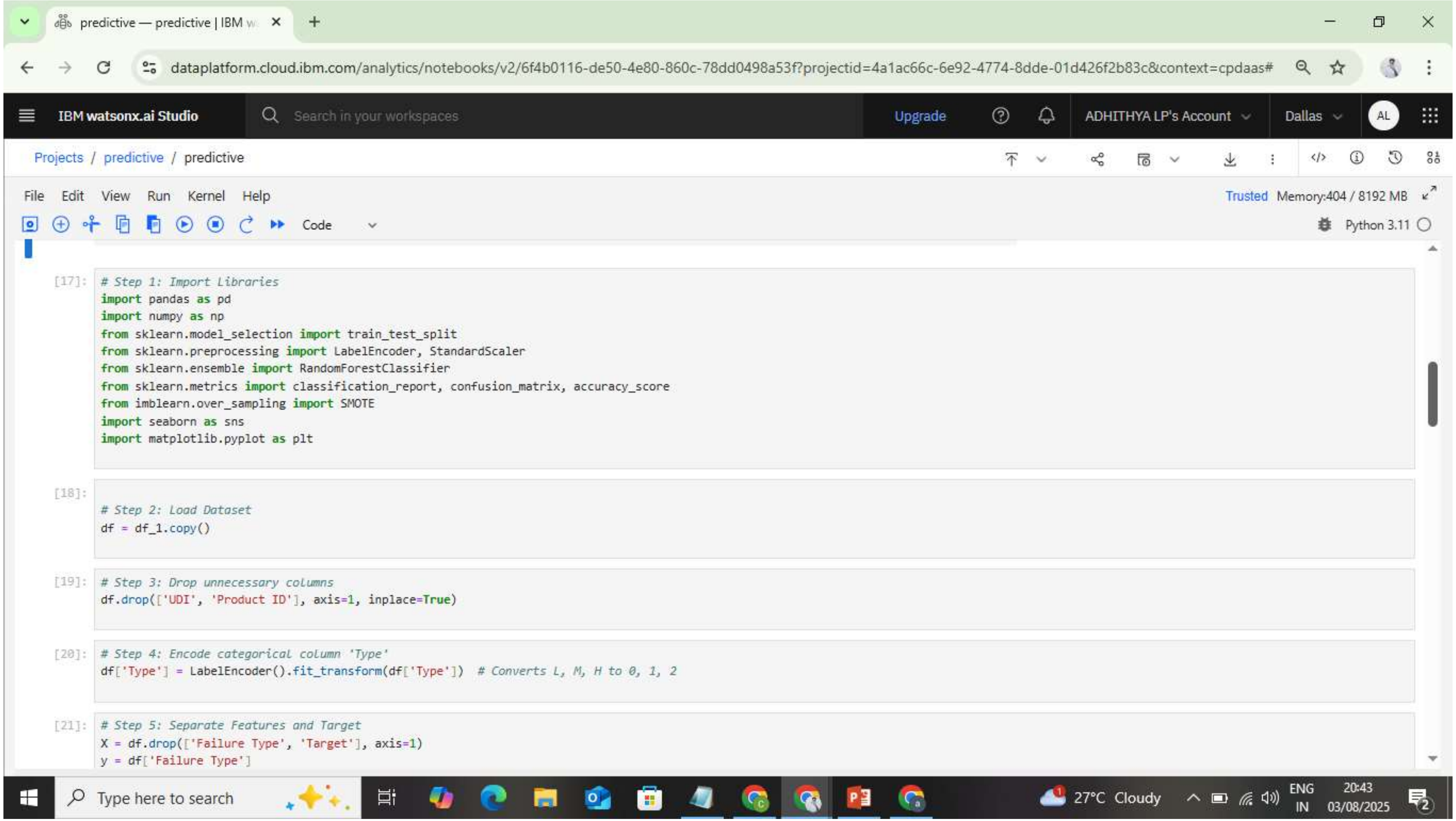

predictive — predictive | IBM
dataplatform.cloud.ibm.com/analytics/notebooks/v2/6f4b0116-de50-4e80-860c-78dd0498a53f?projectid=4a1ac66c-6e92-4774-8dde-01d426f2b83c&context=cpdaas#
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
ADHITHYA LP's Account
Dallas
AL
Projects / predictive / predictive
File Edit View Run Kernel Help
Trusted Memory:404 / 8192 MB
Code
Python 3.11
[1]:
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
def __iter__(self): return 0
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
ibm_api_key_id='6pfkzW97FNMd5GSqdgmSUzcBLU1fwxmf1zzEtae8s_H2',
ibm_auth_endpoint="https://iam.cloud.ibm.com/identity/token",
config=Config(signature_version='oauth'),
endpoint_url='https://s3.direct.us-south.cloud-object-storage.appdomain.cloud')
bucket = 'predictive-donotdelete-pr-shj4qjiy2tuxhv'
object_key = 'predictive_maintenance.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body)
df_1 = pd.read_csv(body)
df_1.head(10)
Type here to search
27°C Cloudy
ENG
IN
20:42
03/08/2025

RESULTS

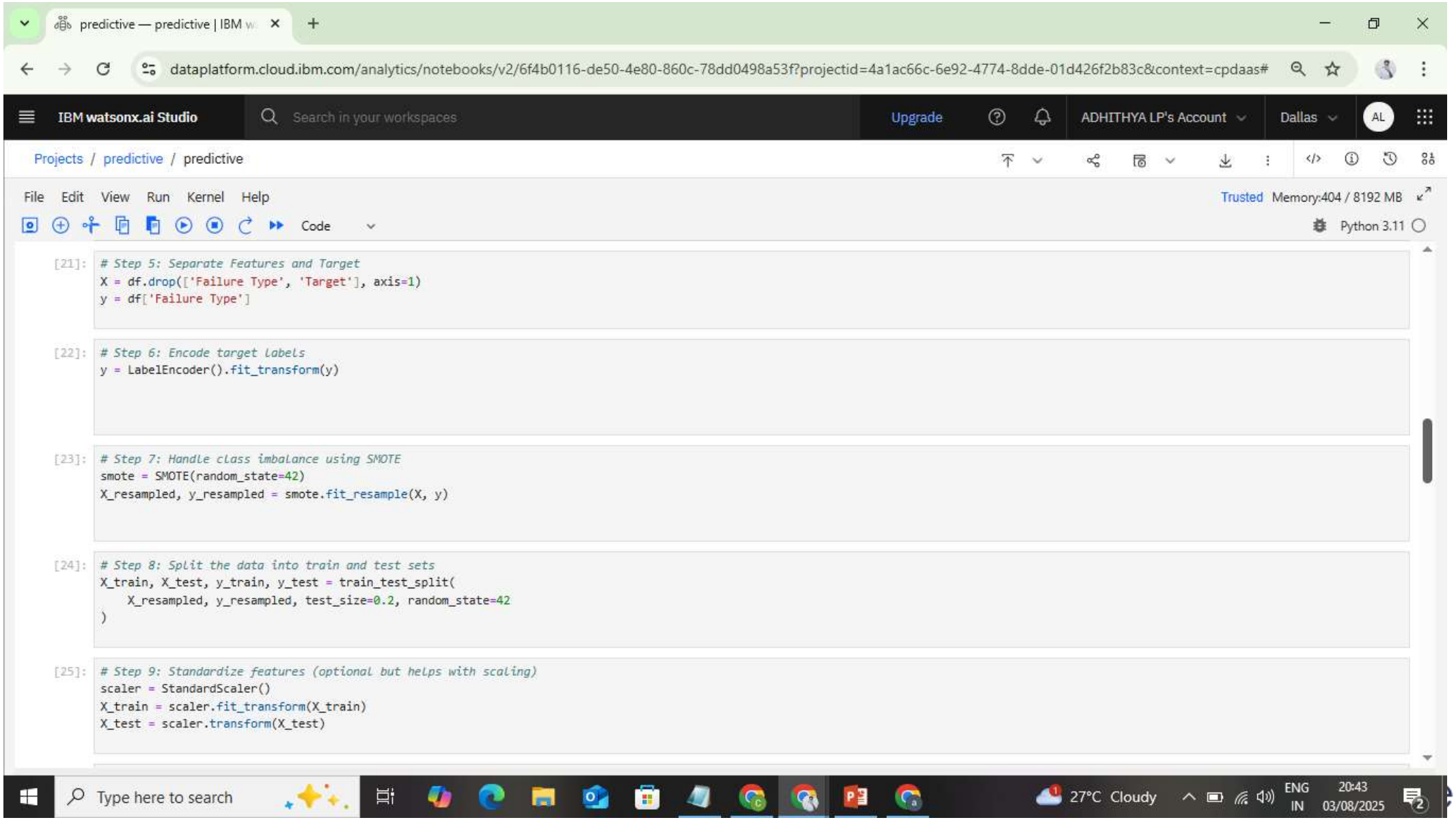


	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	No Failure
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	No Failure
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	No Failure
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	No Failure
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	No Failure
5	6	M14865	M	298.1	308.6	1425	41.9	11	0	No Failure
6	7	L47186	L	298.1	308.6	1558	42.4	14	0	No Failure
7	8	L47187	L	298.1	308.6	1527	40.2	16	0	No Failure
8	9	M14868	M	298.3	308.7	1667	28.6	18	0	No Failure
9	10	M14869	M	298.5	309.0	1741	28.0	21	0	No Failure

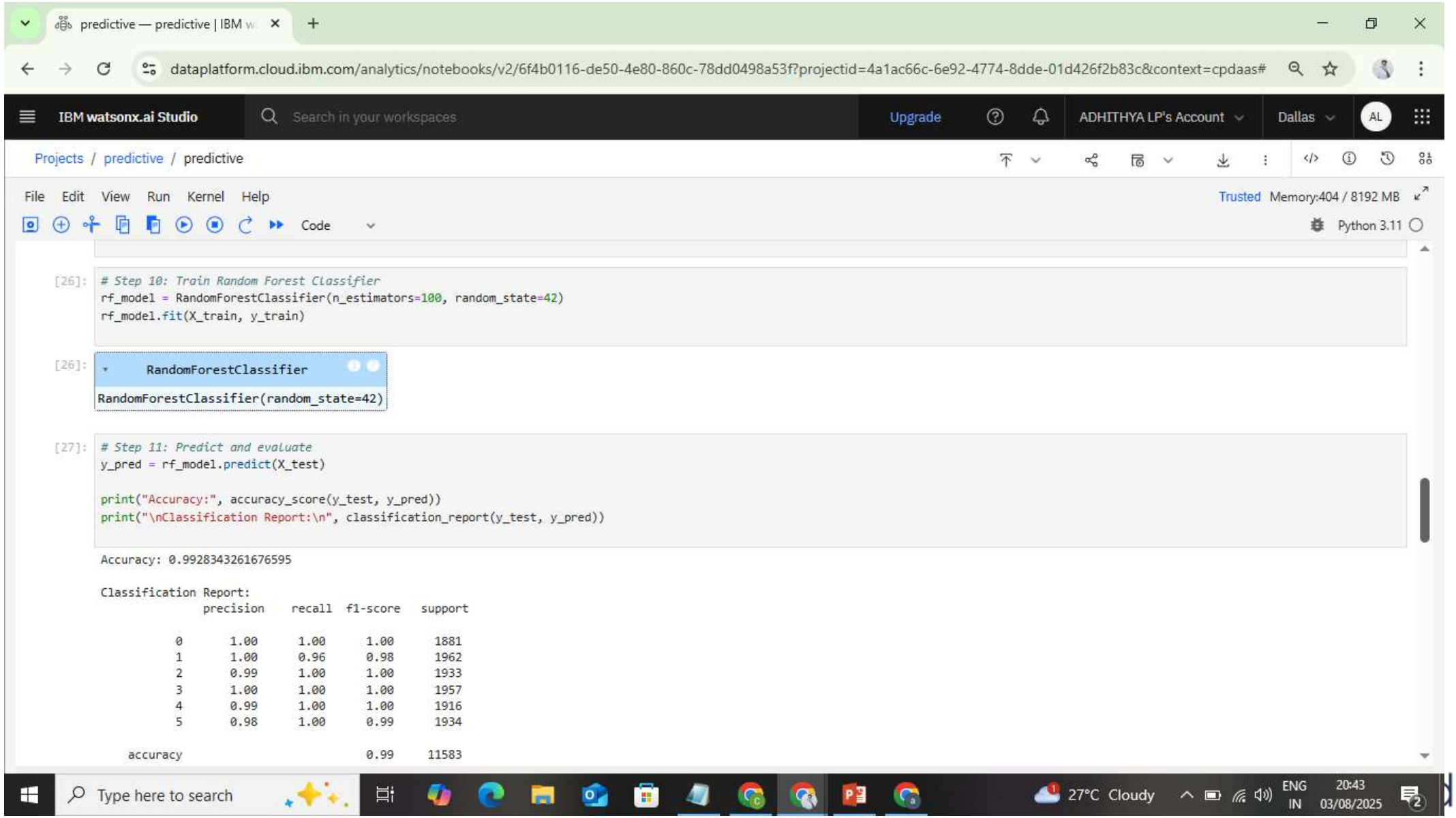
RESULTS


predictive — predictive | IBM w
dataplatform.cloud.ibm.com/analytics/notebooks/v2/6f4b0116-de50-4e80-860c-78dd0498a53f?projectid=4a1ac66c-6e92-4774-8dde-01d426f2b83c&context=cpdaas#
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
ADHITHYA LP's Account
Dallas
AL
Projects / predictive / predictive
File Edit View Run Kernel Help
Trusted Memory:404 / 8192 MB
Code
Python 3.11
[17]: # Step 1: Import Libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from imblearn.over_sampling import SMOTE
import seaborn as sns
import matplotlib.pyplot as plt
[18]: # Step 2: Load Dataset
df = df_1.copy()
[19]: # Step 3: Drop unnecessary columns
df.drop(['UDI', 'Product ID'], axis=1, inplace=True)
[20]: # Step 4: Encode categorical column 'Type'
df['Type'] = LabelEncoder().fit_transform(df['Type']) # Converts L, M, H to 0, 1, 2
[21]: # Step 5: Separate Features and Target
X = df.drop(['Failure Type', 'Target'], axis=1)
y = df['Failure Type']
Type here to search
27°C Cloudy
ENG IN
20:43 03/08/2025

RESULTS



RESULTS


predictive — predictive | IBM
dataplatform.cloud.ibm.com/analytics/notebooks/v2/6f4b0116-de50-4e80-860c-78dd0498a53f?projectid=4a1ac66c-6e92-4774-8dde-01d426f2b83c&context=cpdaas#
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
ADHITHYA LP's Account
Dallas
AL
Projects / predictive / predictive
File Edit View Run Kernel Help
Trusted Memory:404 / 8192 MB
Code
Python 3.11
[26]:
# Step 10: Train Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
[26]:
RandomForestClassifier
RandomForestClassifier(random_state=42)
[27]:
# Step 11: Predict and evaluate
y_pred = rf_model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
Accuracy: 0.9928343261676595
Classification Report:
precision recall f1-score support
0 1.00 1.00 1.00 1881
1 1.00 0.96 0.98 1962
2 0.99 1.00 1.00 1933
3 1.00 1.00 1.00 1957
4 0.99 1.00 1.00 1916
5 0.98 1.00 0.99 1934
accuracy 0.99 11583
Type here to search
27°C Cloudy
ENG IN
20:43 03/08/2025

RESULTS

predictive — predictive | IBM w

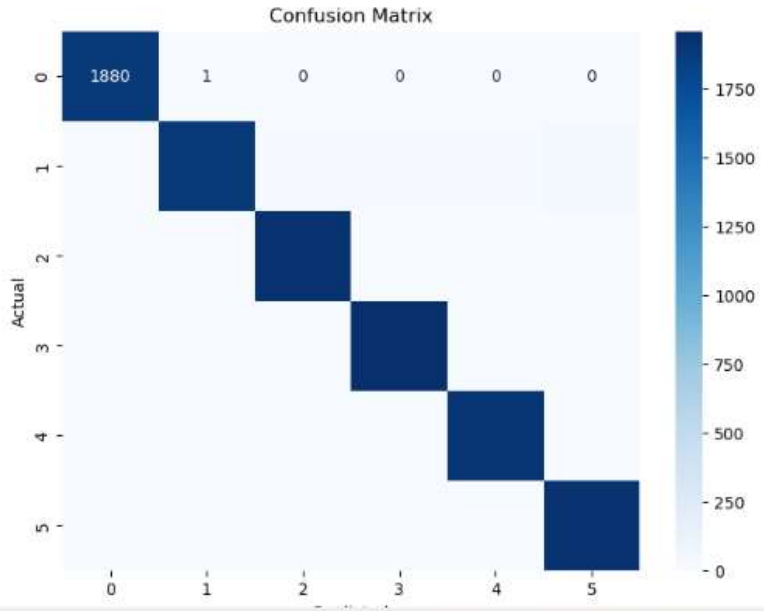
dataplatform.cloud.ibm.com/analytics/notebooks/v2/6f4b0116-de50-4e80-860c-78dd0498a53f?projectid=4a1ac66c-6e92-4774-8dde-01d426f2b83c&context=cpdaas#

IBM watsonx.ai Studio | Search in your workspaces | Upgrade | ADHITHYA LP's Account | Dallas | AL

Projects / predictive / predictive

File Edit View Run Kernel Help | Trusted | Memory:404 / 8192 MB | Code | Python 3.11

[28]:
```
# Step 12: Confusion Matrix
plt.figure(figsize=(8,6))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



Type here to search | 27°C Cloudy | ENG IN | 20:44 03/08/2025

edunet foundation

RESULTS

predictive — predictive | IBM

dataplatform.cloud.ibm.com/analytics/notebooks/v2/6f4b0116-de50-4e80-860c-78dd0498a53f?projectid=4a1ac66c-6e92-4774-8dde-01d426f2b83c&context=cpdaas#

IBM watsonx.ai Studio

Projects / predictive / predictive

File Edit View Run Kernel Help

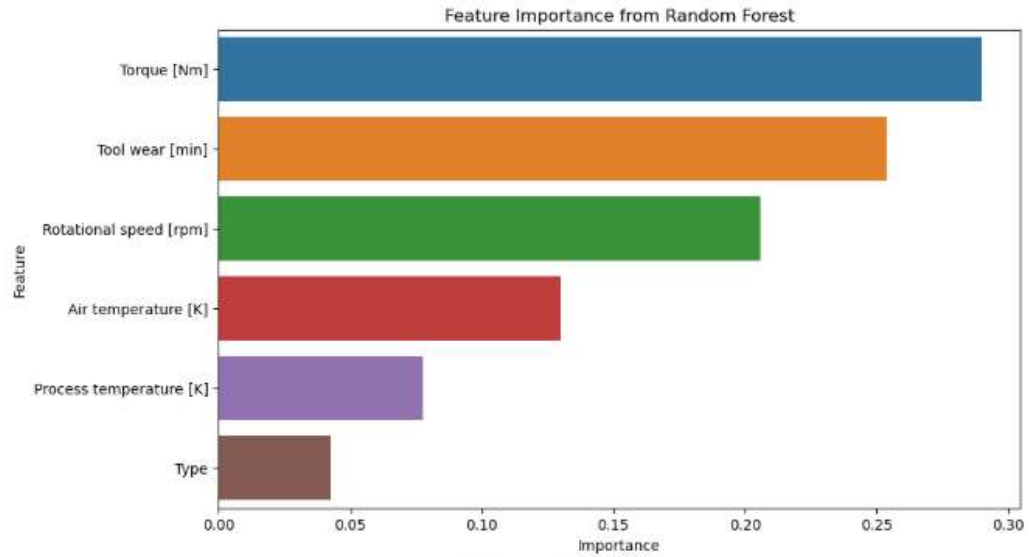
Trusted Memory:404 / 8192 MB

Code

Python 3.11

```
[29]: # Step 13: Feature Importance
importances = rf_model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({'Feature': features, 'Importance': importances})
importance_df.sort_values(by='Importance', ascending=False, inplace=True)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df)
plt.title('Feature Importance from Random Forest')
plt.show()
```



Type here to search

27°C Cloudy

ENG IN

20:45 03/08/2025

dunet foundation

CONCLUSION

- Developing a high performing predictive maintenance model that accurately classifies different machine failure types using real-time sensor data.
- Significantly reduces unplanned downtime and maintenance cost be enabling proactive decision making.
- The model is scalable and ready for deployment on cloud platforms like IBM cloud for real-world industrial applications.

FUTURE SCOPE

- Real-time monitoring integration
- Model deployment via API
- Advanced deep learning models
- Predictive analytics dashboard
- Automated Maintenance alerts
- Cross-Domain Application

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

ADHITHYA LP

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

edunet foundation

GITHUB LINK

- Github link: https://github.com/Adhi-1235/Predictive-Maintenance.git

THANK YOU

edunet foundation