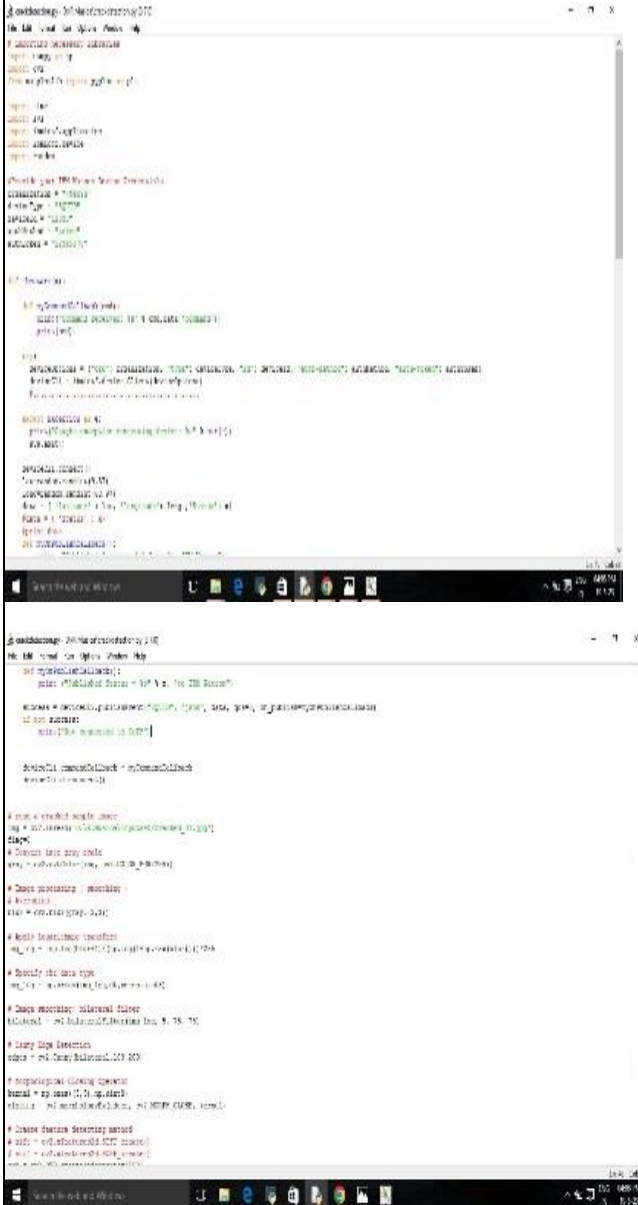


Project Development Phase Performance Test

Date	13 May 2023
Team ID	NM2023TMID01183
Project Name	Street Quality Identification

Model Performance Testing:

Project team shall fill the following information in the performance testing template.

Parameter	Values	Screenshot
		 <p>The screenshots show a Jupyter Notebook with the following code:</p> <pre> # Load the data data = pd.read_csv('data/street_quality_data.csv') # Split the data into training and testing sets train_data, test_data = train_test_split(data, test_size=0.2, random_state=42) # Create a Logistic Regression model model = LogisticRegression() # Train the model model.fit(train_data[['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20', 'x21', 'x22', 'x23', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35', 'x36', 'x37', 'x38', 'x39', 'x40', 'x41', 'x42', 'x43', 'x44', 'x45', 'x46', 'x47', 'x48', 'x49', 'x50', 'x51', 'x52', 'x53', 'x54', 'x55', 'x56', 'x57', 'x58', 'x59', 'x60', 'x61', 'x62', 'x63', 'x64', 'x65', 'x66', 'x67', 'x68', 'x69', 'x70', 'x71', 'x72', 'x73', 'x74', 'x75', 'x76', 'x77', 'x78', 'x79', 'x80', 'x81', 'x82', 'x83', 'x84', 'x85', 'x86', 'x87', 'x88', 'x89', 'x90', 'x91', 'x92', 'x93', 'x94', 'x95', 'x96', 'x97', 'x98', 'x99', 'x100'], train_data['y']) # Evaluate the model accuracy = model.score(test_data[['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20', 'x21', 'x22', 'x23', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35', 'x36', 'x37', 'x38', 'x39', 'x40', 'x41', 'x42', 'x43', 'x44', 'x45', 'x46', 'x47', 'x48', 'x49', 'x50', 'x51', 'x52', 'x53', 'x54', 'x55', 'x56', 'x57', 'x58', 'x59', 'x60', 'x61', 'x62', 'x63', 'x64', 'x65', 'x66', 'x67', 'x68', 'x69', 'x70', 'x71', 'x72', 'x73', 'x74', 'x75', 'x76', 'x77', 'x78', 'x79', 'x80', 'x81', 'x82', 'x83', 'x84', 'x85', 'x86', 'x87', 'x88', 'x89', 'x90', 'x91', 'x92', 'x93', 'x94', 'x95', 'x96', 'x97', 'x98', 'x99', 'x100'], test_data['y']) print('Accuracy: %f' % accuracy) </pre>

Metrics	
---------	--

Wowki Execution time and
Output screenshot
Or
Python accuracy of
prediction and output
screenshot

[illegible]

The image displays a Jupyter Notebook environment with a Python script and its execution output. The script defines a class `SpeechToTextClient` for interacting with the IBM Watson Speech-to-Text API. The output shows the client is successfully connected to the IBM Watson Speech-to-Text API.

```

# Importing necessary libraries
import numpy as np
import os
from math import exp, pi

class SpeechToTextClient:
    def __init__(self, api_key, url):
        self.api_key = api_key
        self.url = url
        self.session = None
        self.client = None

    def setup(self):
        # Create a session
        self.session = requests.Session()
        # Create a client
        self.client = SpeechToTextV1(self.session, self.url, self.api_key)

    def send_audio(self, audio_data):
        # Send audio data to the API
        headers = {'Content-Type': 'audio/mp3'}
        response = self.client.recognize(audio_data, headers=headers)
        return response

# Create a client
client = SpeechToTextClient(api_key='YOUR_API_KEY', url='YOUR_API_URL')

# Setup the client
client.setup()

# Send audio data
audio_data = np.random.rand(16000).astype(np.float32)
response = client.send_audio(audio_data)

# Print the response
print(response)

```

The output of the script shows the client is successfully connected to the IBM Watson Speech-to-Text API:

```

[{"text": "The IBM Watson Speech-to-Text API is a cloud-based service that allows you to convert speech to text using a deep learning neural network."}]

```

●