Media Monitoring Multilabel Classification:Multilabel Classification Of Printed Media Articles To Topics

INTRODUCTION:

The purpose of this project is to develop a multilabel classification system for printed media articles. The system will be capable of assigning multiple labels or topics to each article, enabling efficient categorization and analysis of media content. This documentation outlines the objectives, methodology, and implementation details of the project.

OVERVIEW:

The main objectives of the project are as follows:

Develop a robust multilabel classification model for printed media articles.

Implement a scalable system capable of handling a large volume of articles.

Improve the accuracy and efficiency of media monitoring and analysis.

PURPOSE:

Enable quick and accurate categorization of articles into multiple topics.

Methodology: The project will follow the following methodology:

Data Collection: Gather a diverse dataset of printed media articles from various sources.

Data Preprocessing: Clean and preprocess the collected data by removing noise, formatting inconsistencies, and irrelevant information.

LITRACHURE SURVEY:

EXSISTING PROBLEM:

Feature Extraction: Extract relevant features from the preprocessed data, such as keywords, entities, and textual representations.

PROPOSED SOLUTION:

Model Development: Train a multilabel classification model using appropriate algorithms, such as deep learning models (e.g., Convolutional Neural Networks, Recurrent Neural Networks) or ensemble methods (e.g., Random Forest, Gradient Boosting).

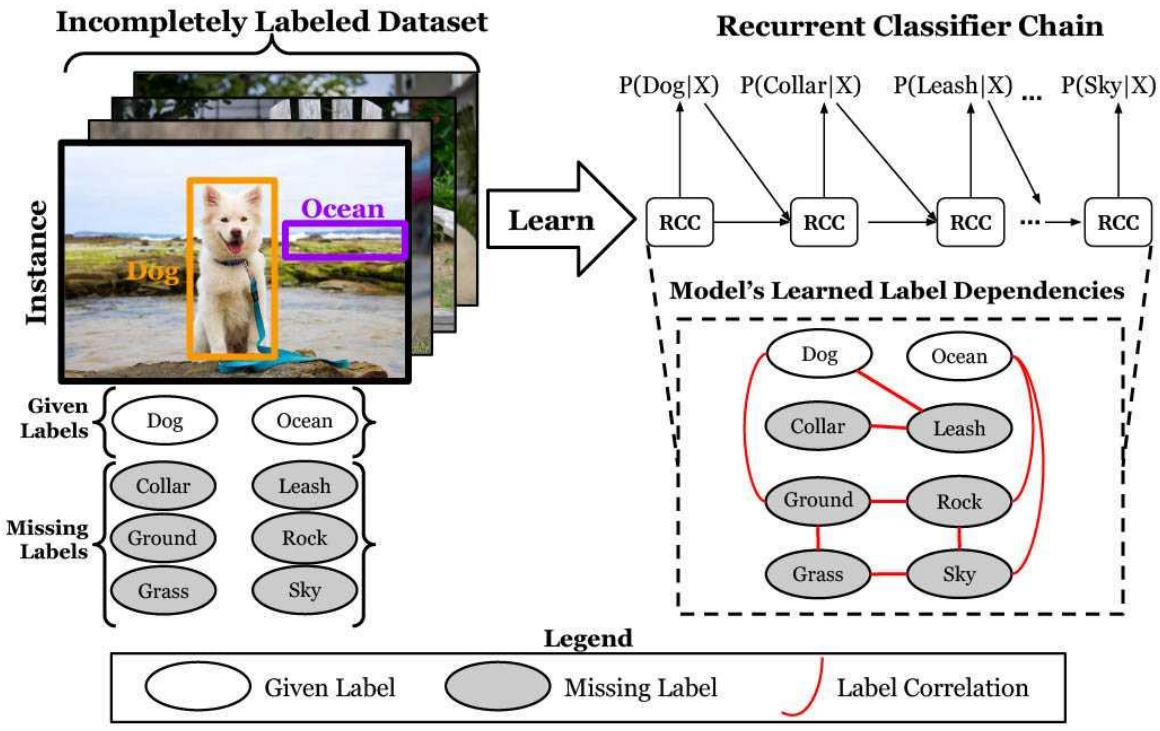
Model Evaluation: Evaluate the performance of the developed model using appropriate metrics, such as precision, recall, and F1-score.

Model Optimization: Fine-tune the model parameters and hyperparameters to improve its performance.

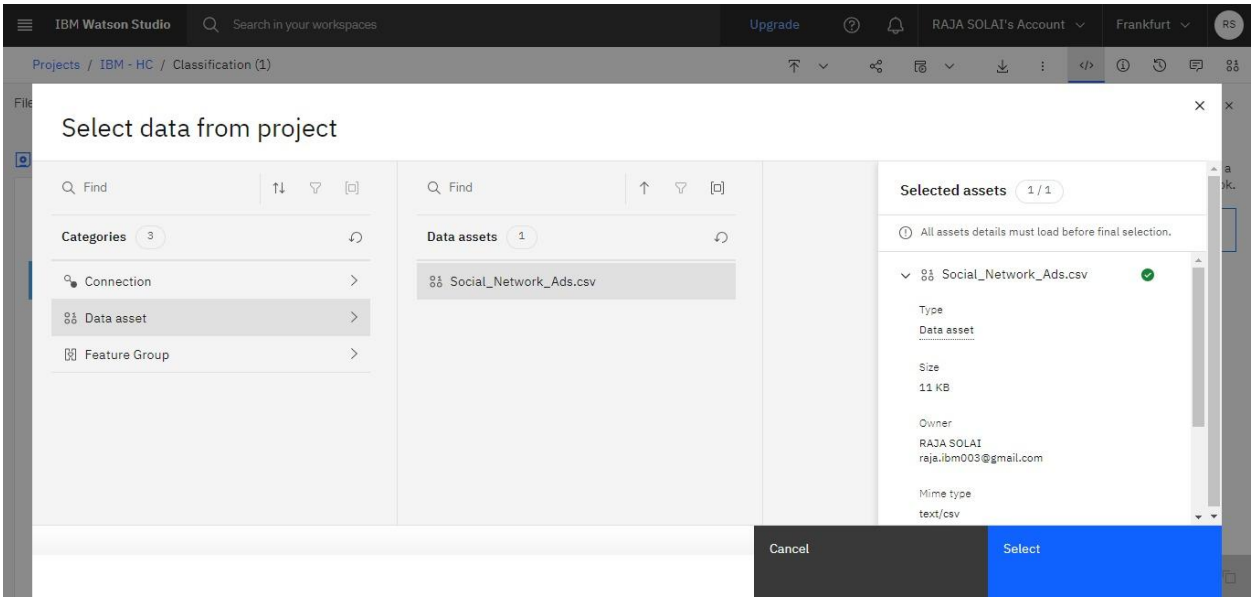
Deployment: Implement the classification system in a scalable and efficient manner, allowing for real-time or batch processing of articles.

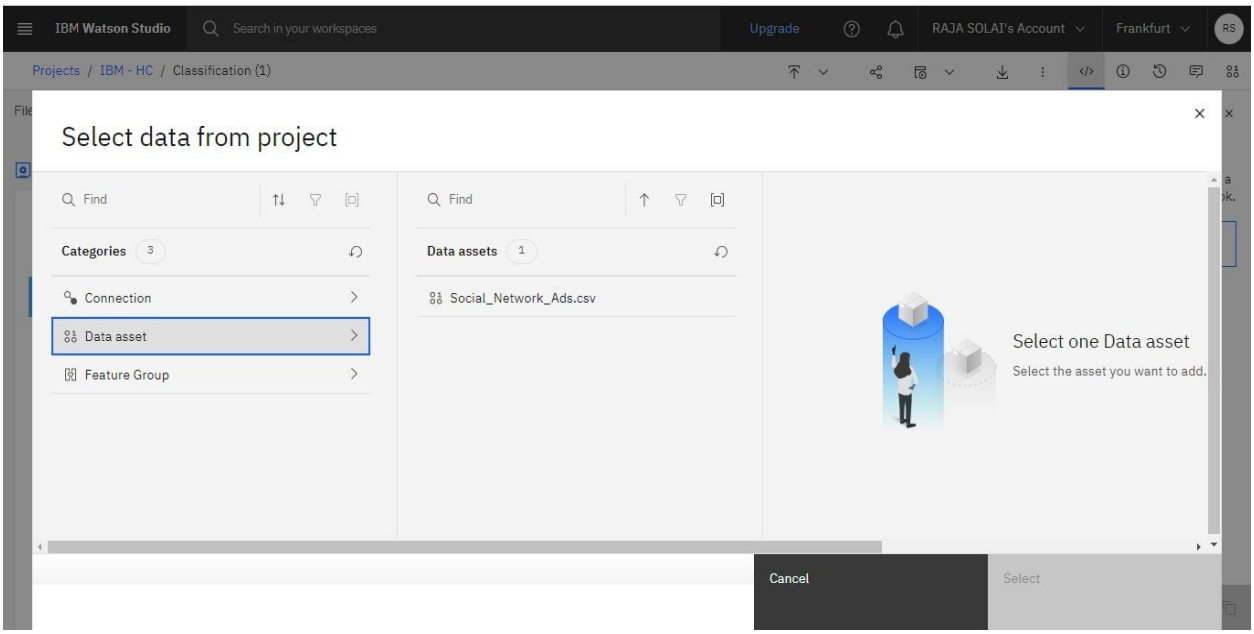
Testing and Validation: Test the system using a separate validation dataset to ensure its accuracy and reliability.

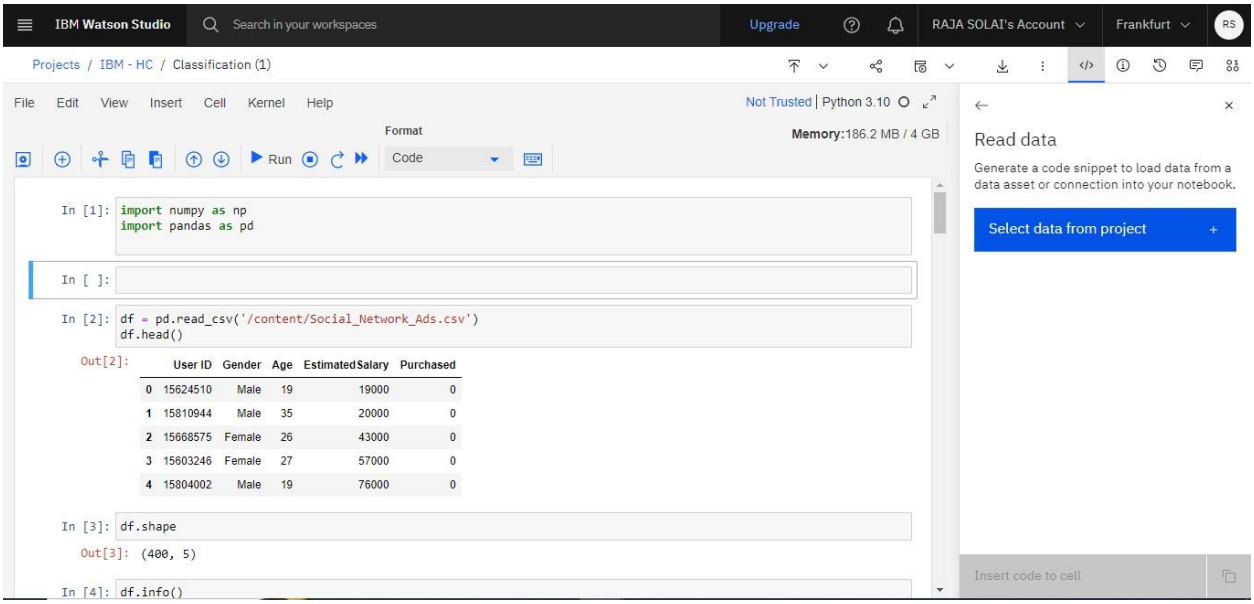
FLOWCHART:

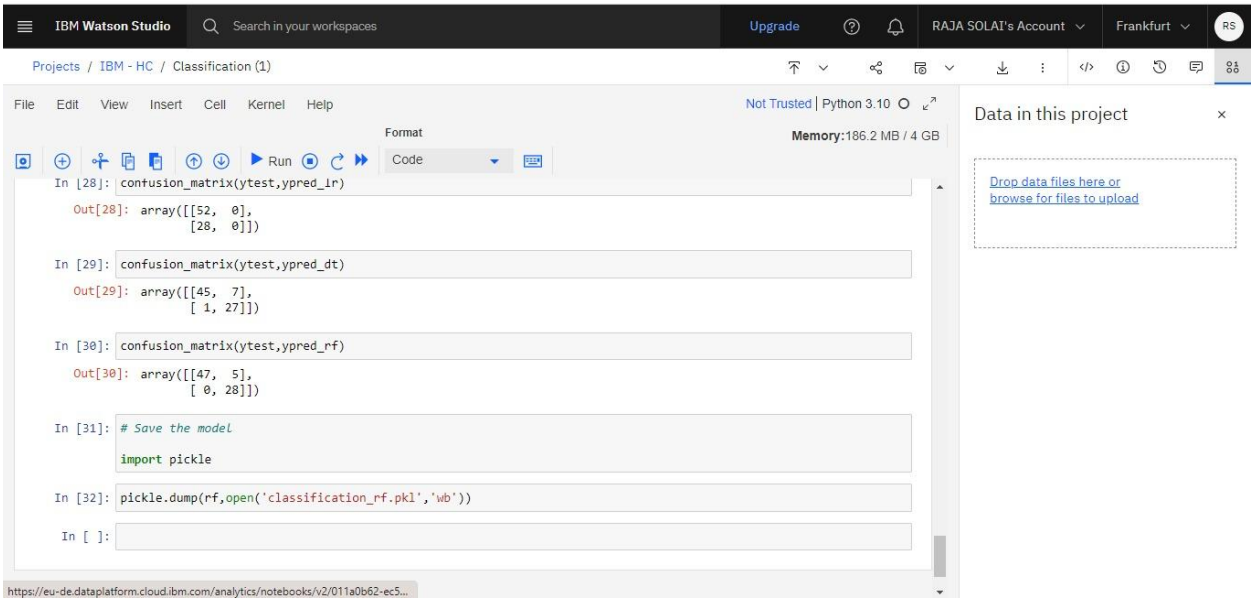


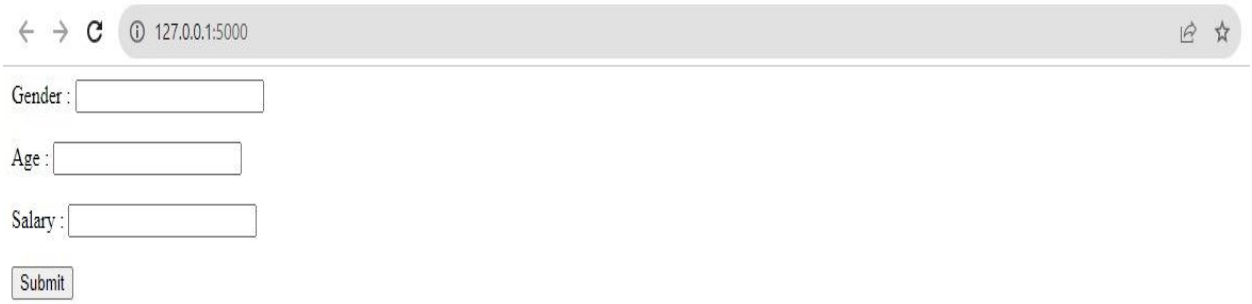
Result:

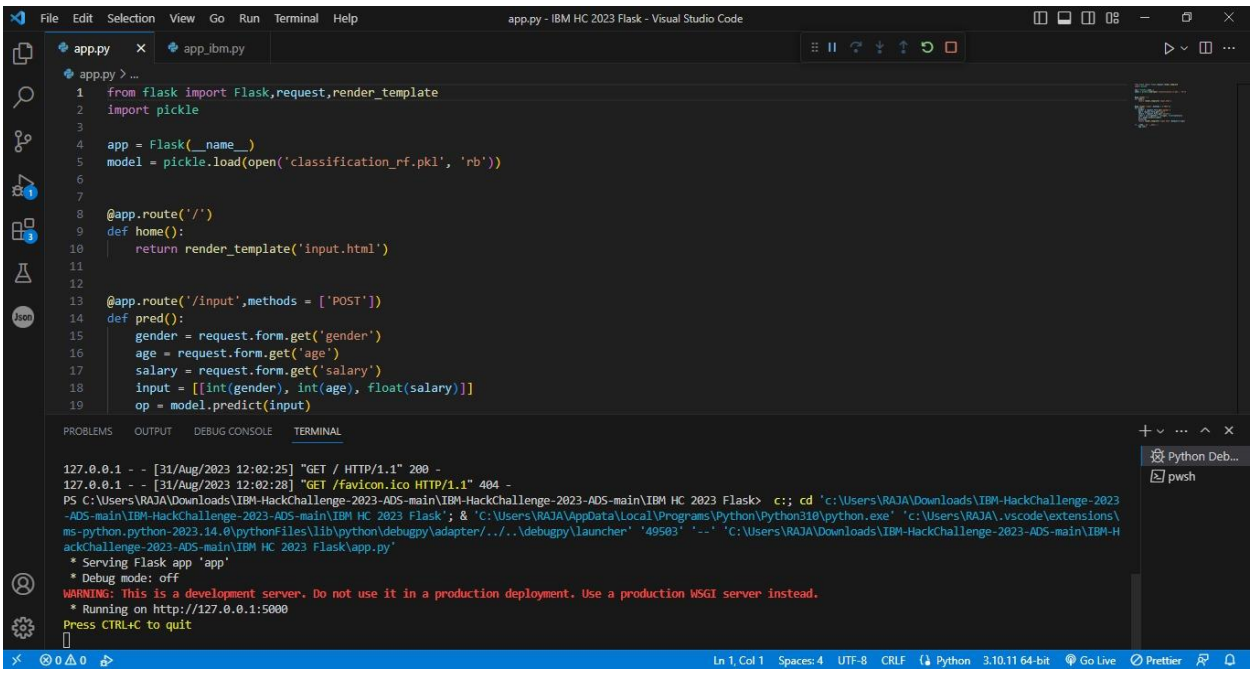

IBM Watson Studio
Search in your workspaces
Upgrade
RAJA SOLAI's Account
Frankfurt
Projects / IBM - HC / Classification (1)
Select data from project
Find
Categories 3
Connection
Data asset
Feature Group
Find
Data assets 1
Social_Network_Ads.csv
Selected assets 1 / 1
All assets details must load before final selection.
Social_Network_Ads.csv
Type
Data asset
Size
11 KB
Owner
RAJA SOLAI
raja.ibm003@gmail.com
Mime type
text/csv
Cancel
Select

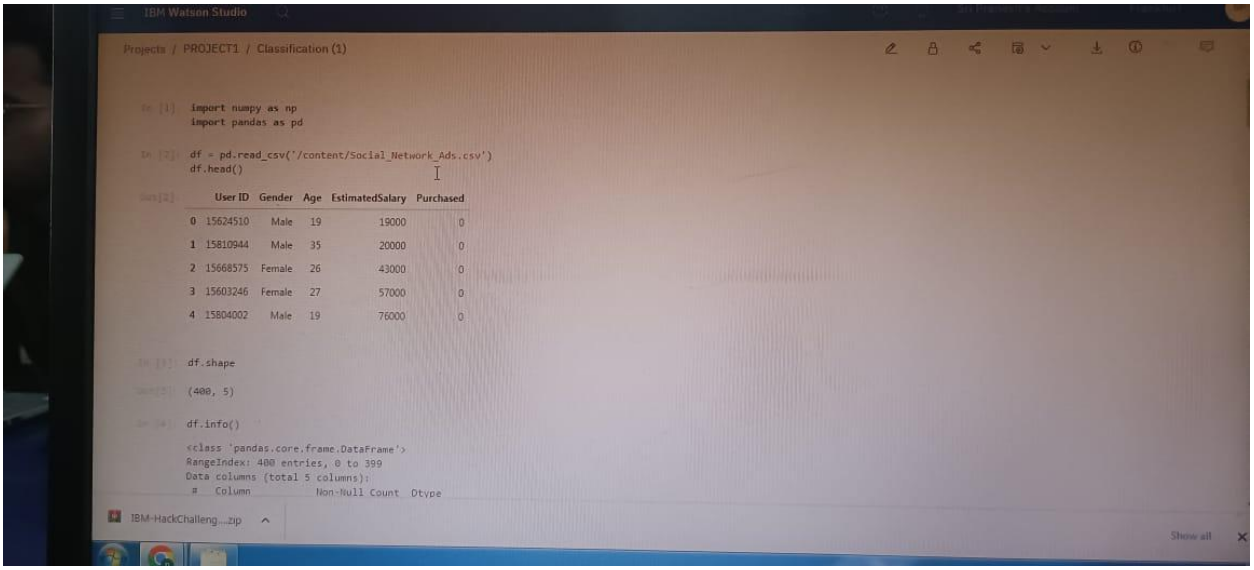

IBM Watson Studio
Search in your workspaces
Upgrade
RAJA SOLAI's Account
Frankfurt
Projects / IBM - HC / Classification (1)
Select data from project
Find
Categories 3
Connection
Data asset
Feature Group
Find
Data assets 1
Social_Network_Ads.csv
Select one Data asset
Select the asset you want to add.
Cancel
Select

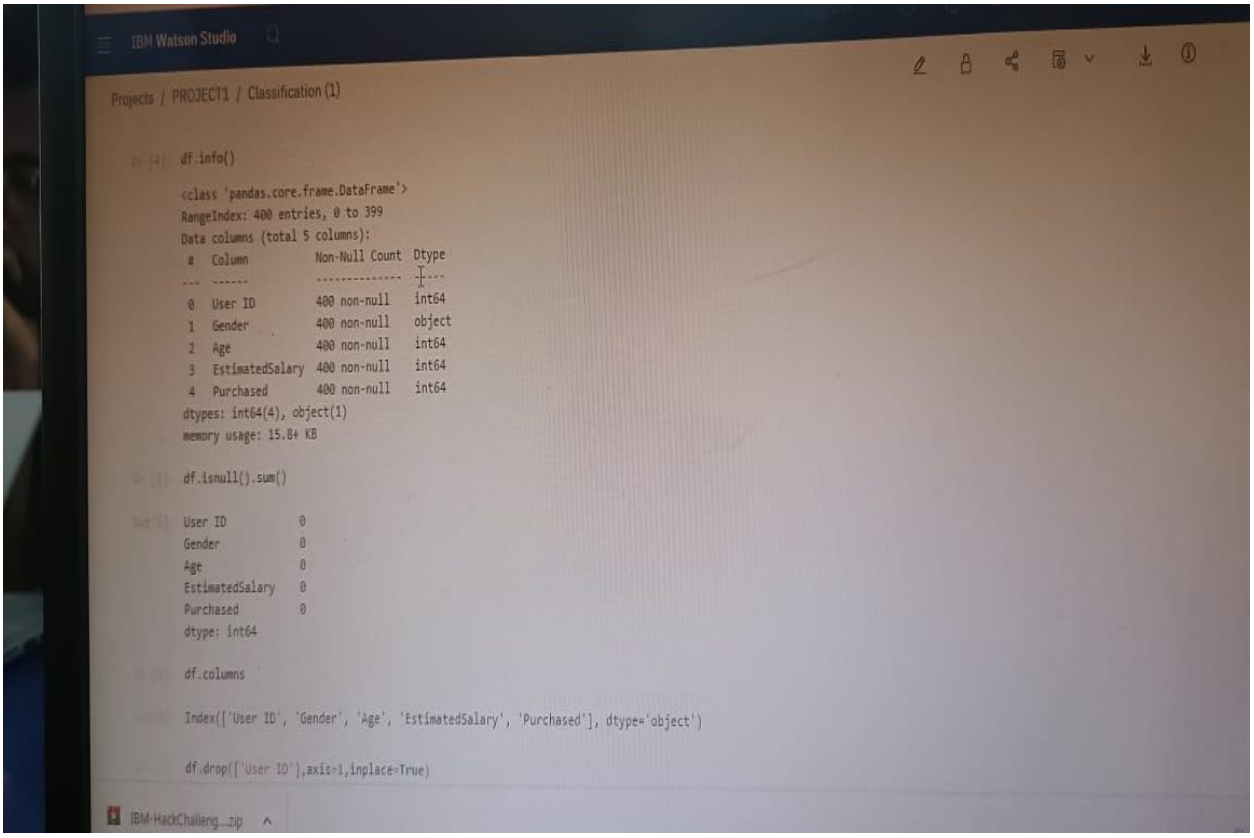

IBM Watson Studio
Search in your workspaces
Upgrade
RAJA SOLAI's Account
Frankfurt
RS
Projects / IBM - HC / Classification (1)
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.10
Memory:186.2 MB / 4 GB
Format
Run
Code
In [1]: import numpy as np
import pandas as pd
In [ ]:
In [2]: df = pd.read_csv('/content/Social_Network_Ads.csv')
df.head()
Out[2]:
User ID Gender Age EstimatedSalary Purchased
0 15624510 Male 19 19000 0
1 15810944 Male 35 20000 0
2 15668575 Female 26 43000 0
3 15603246 Female 27 57000 0
4 15804002 Male 19 76000 0
In [3]: df.shape
Out[3]: (400, 5)
In [4]: df.info()
Read data
Generate a code snippet to load data from a data asset or connection into your notebook.
Select data from project
Insert code to cell

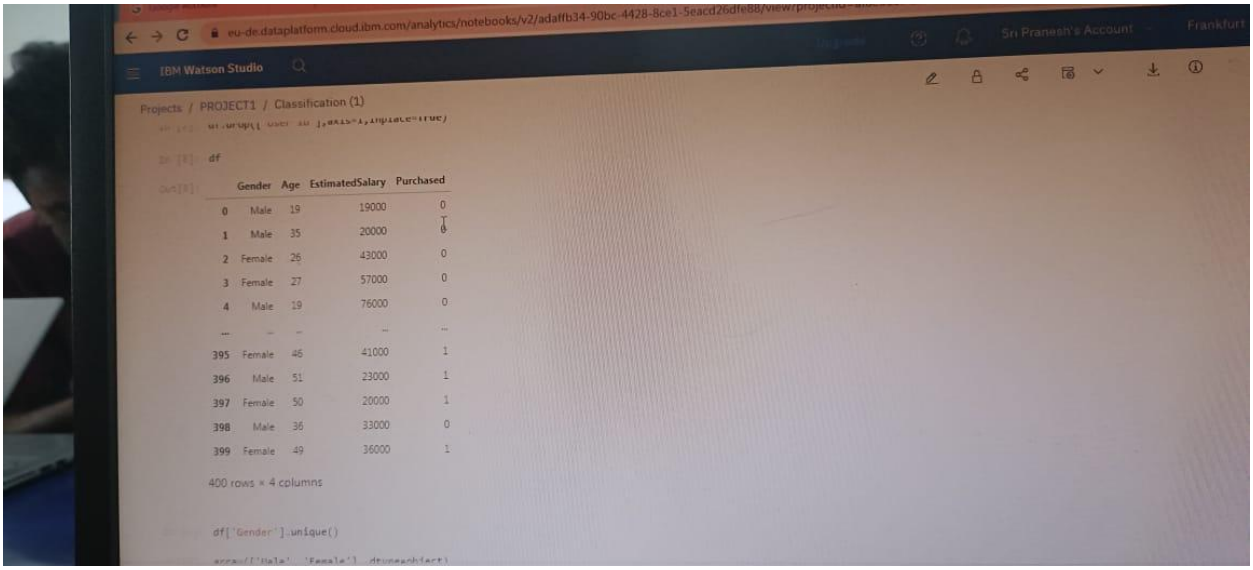

IBM Watson Studio
Search in your workspaces
Upgrade
RAJA SOLAI's Account
Frankfurt
RS
Projects / IBM - HC / Classification (1)
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.10
Memory:186.2 MB / 4 GB
Format
Run
Code
In [28]: confusion_matrix(ytest,ypred_lr)
Out[28]: array([[52, 0],
[28, 0]])
In [29]: confusion_matrix(ytest,ypred_dt)
Out[29]: array([[45, 7],
[1, 27]])
In [30]: confusion_matrix(ytest,ypred_rf)
Out[30]: array([[47, 5],
[0, 28]])
In [31]: # Save the model
import pickle
In [32]: pickle.dump(rf,open('classification_rf.pkl','wb'))
In [ ]:
Data in this project
Drop data files here or browse for files to upload
https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/011a0b62-ec5...


127.0.0.1:5000
Gender :
Age :
Salary :
Submit
Output :


File Edit Selection View Go Run Terminal Help
app.py - IBM HC 2023 Flask - Visual Studio Code
app.py
app_ibm.py
app.py > ...
1 from flask import Flask,request,render_template
2 import pickle
3
4 app = Flask(__name__)
5 model = pickle.load(open('classification_rf.pkl', 'rb'))
6
7
8 @app.route('/')
9 def home():
10 return render_template('input.html')
11
12
13 @app.route('/input',methods = ['POST'])
14 def pred():
15 gender = request.form.get('gender')
16 age = request.form.get('age')
17 salary = request.form.get('salary')
18 input = [[int(gender), int(age), float(salary)]]
19 op = model.predict(input)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Deb...
pwsh
127.0.0.1 - - [31/Aug/2023 12:02:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2023 12:02:28] "GET /favicon.ico HTTP/1.1" 404 -
PS C:\Users\RAJA\Downloads\IBM-HackChallenge-2023-ADS-main\IBM-HackChallenge-2023-ADS-main\IBM HC 2023 Flask> c:; cd 'c:\Users\RAJA\Downloads\IBM-HackChallenge-2023-ADS-main\IBM-HackChallenge-2023-ADS-main\IBM HC 2023 Flask'; & 'C:\Users\RAJA\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\RAJA\.vscode\extensions\ms-python.python-2023.14.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '49503' '--' 'C:\Users\RAJA\Downloads\IBM-HackChallenge-2023-ADS-main\IBM-HackChallenge-2023-ADS-main\IBM HC 2023 Flask\app.py'
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.11 64-bit Go Live Prettier

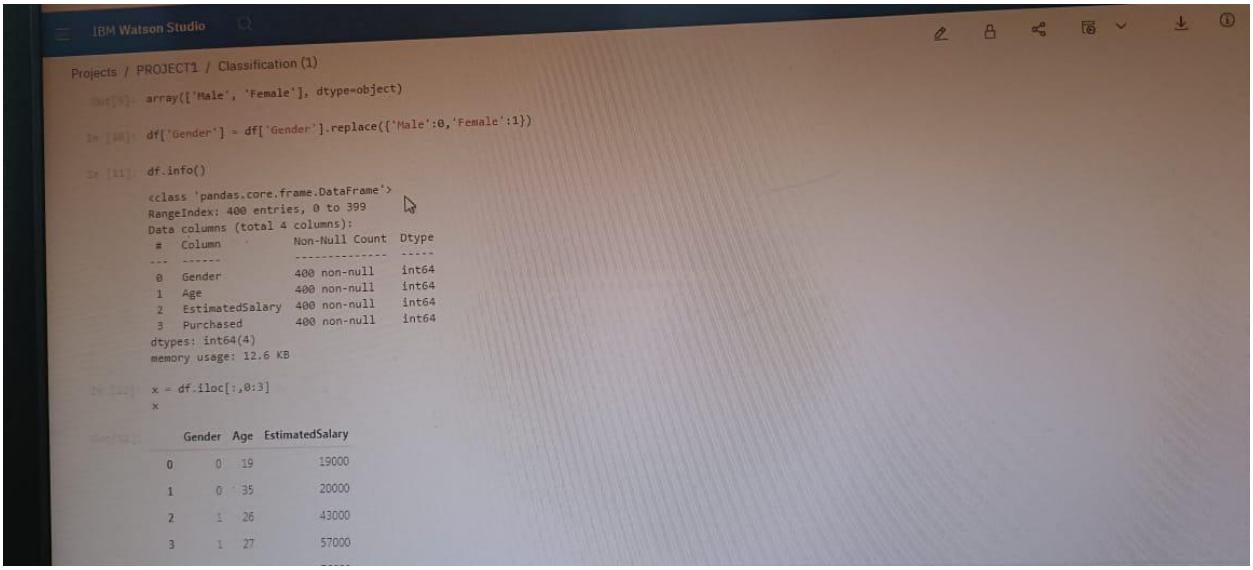

IBM Watson Studio
Projects / PROJECT1 / Classification (1)
import numpy as np
import pandas as pd
df = pd.read_csv('/content/Social_Network_Ads.csv')
df.head()
User ID Gender Age EstimatedSalary Purchased
0 15624510 Male 19 19000 0
1 15810944 Male 35 20000 0
2 15668575 Female 26 43000 0
3 15603246 Female 27 57000 0
4 15804002 Male 19 76000 0
df.shape
(400, 5)
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
Column Non-Null Count Dtype
IBM-HackChalleng....zip
Show all


IBM Watson Studio
Projects / PROJECT1 / Classification (1)
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
Column Non-Null Count Dtype
0 User ID 400 non-null int64
1 Gender 400 non-null object
2 Age 400 non-null int64
3 EstimatedSalary 400 non-null int64
4 Purchased 400 non-null int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
df.isnull().sum()
User ID 0
Gender 0
Age 0
EstimatedSalary 0
Purchased 0
dtype: int64
df.columns
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='object')
df.drop(['User ID'],axis=1,inplace=True)
IBM-HackChalleng....zip

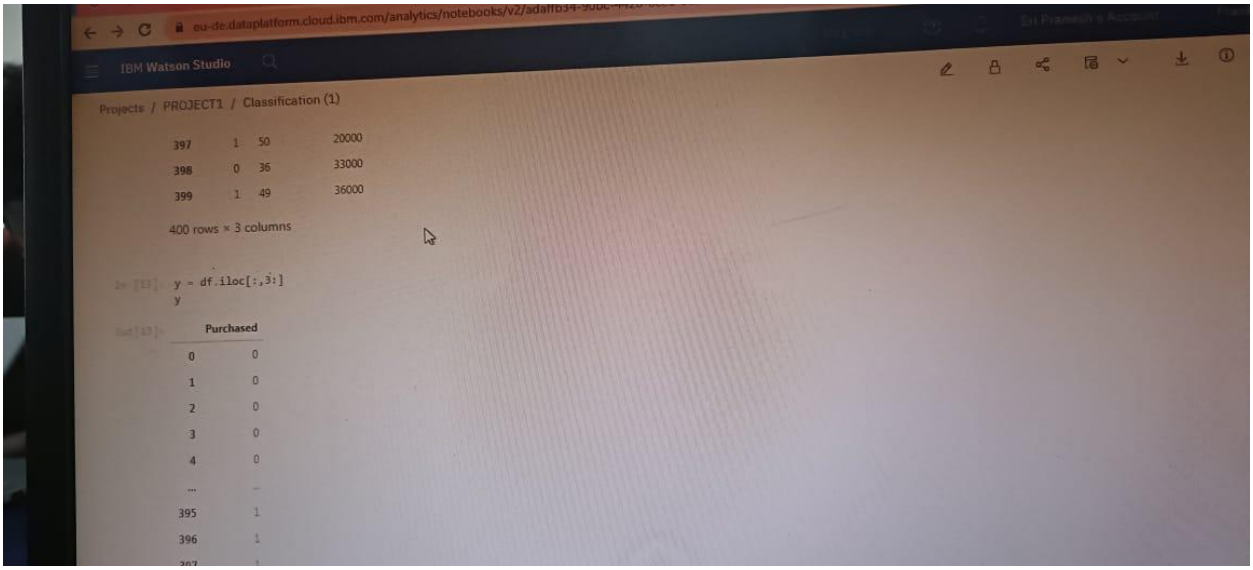


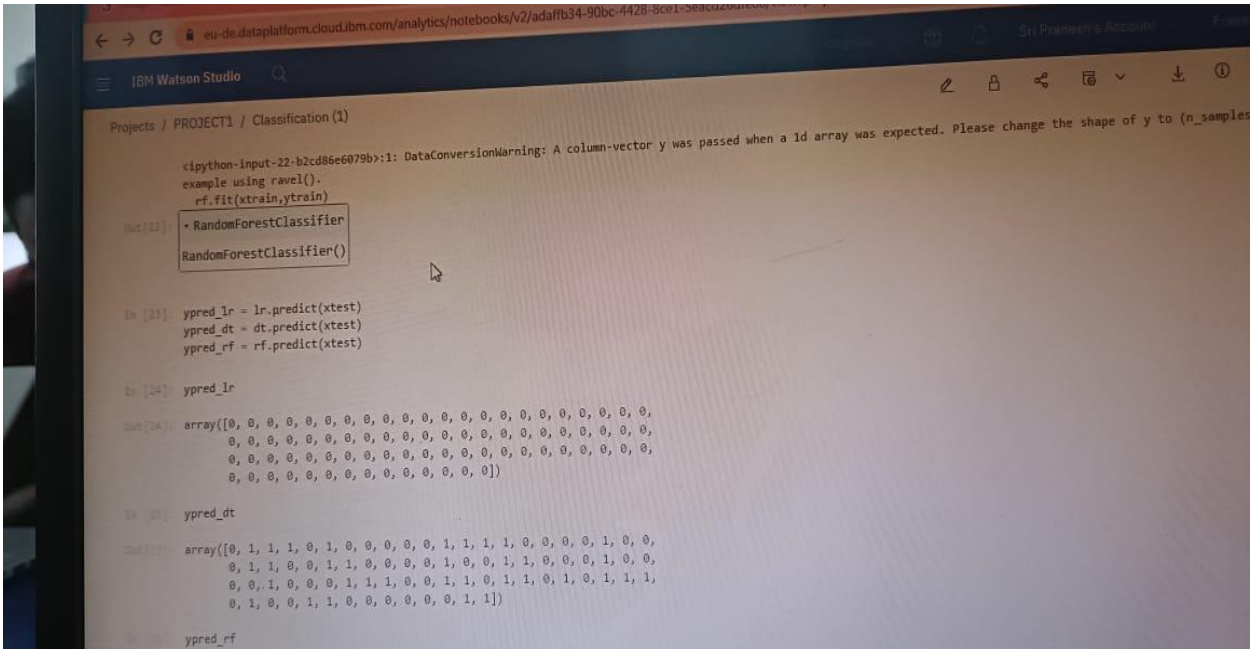
	Gender	Age	EstimatedSalary	Purchased
0	Male	19	19000	0
1	Male	35	20000	0
2	Female	26	43000	0
3	Female	27	57000	0
4	Male	19	76000	0
...	...	...	...	...
395	Female	46	41000	1
396	Male	51	23000	1
397	Female	50	20000	1
398	Male	36	33000	0
399	Female	49	36000	1

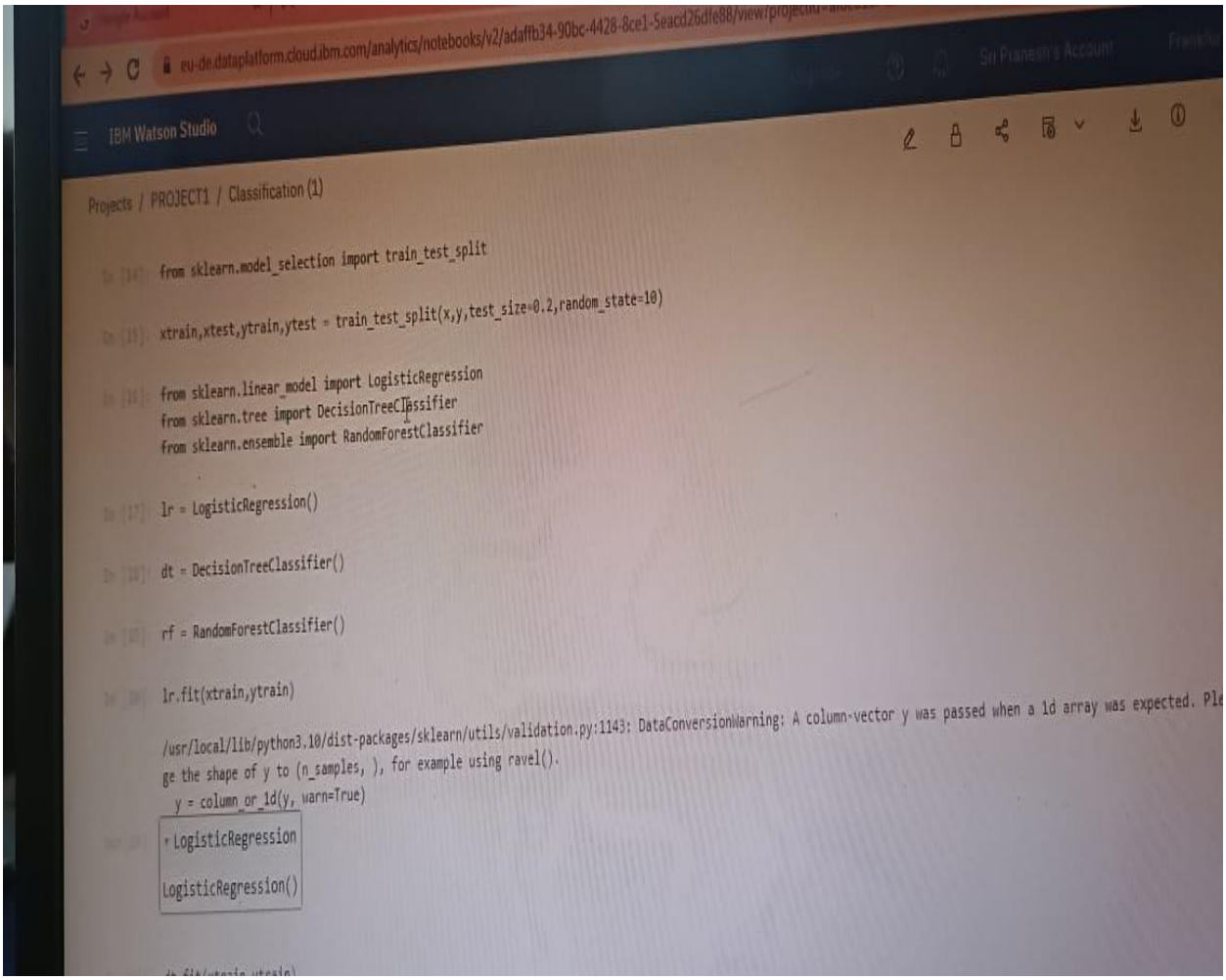
400 rows × 4 columns

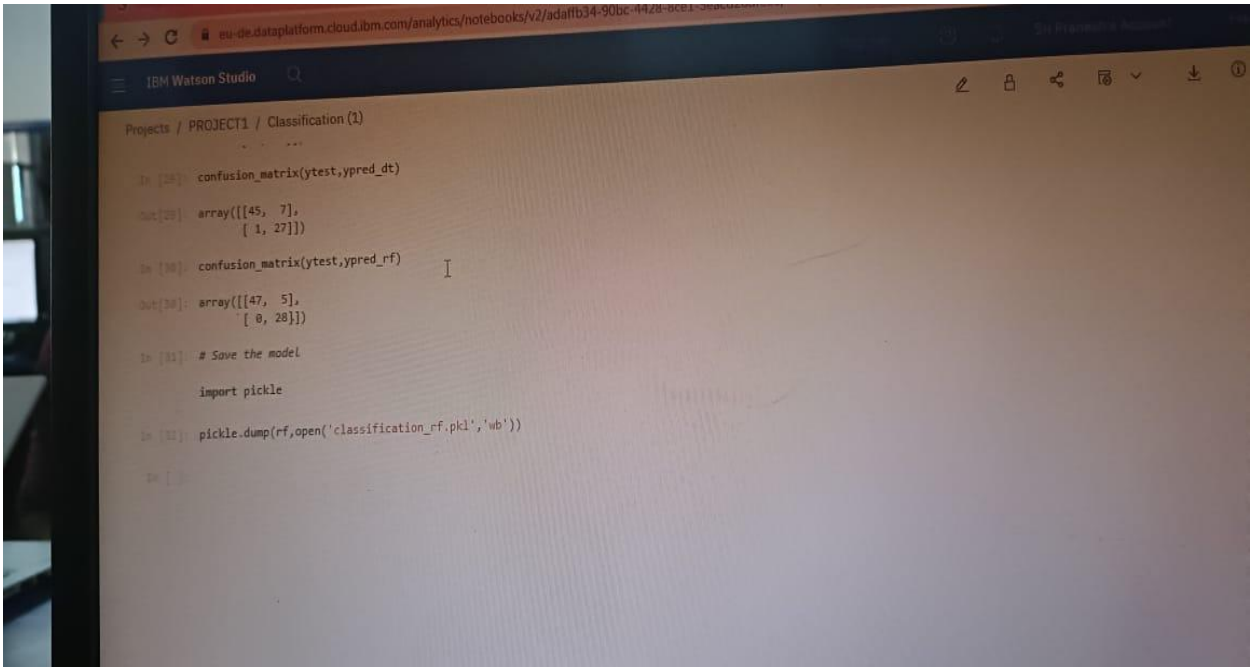


	Gender	Age	EstimatedSalary
0	0	19	19000
1	0	35	20000
2	1	26	43000
3	1	27	57000


IBM Watson Studio
Projects / PROJECT1 / Classification (1)
397 1 50 20000
398 0 36 33000
399 1 49 36000
400 rows × 3 columns
y = df.iloc[:,3:]
y
Purchased
0 0
1 0
2 0
3 0
4 0
395 1
396 1


IBM Watson Studio
Projects / PROJECT1 / Classification (1)
<ipython-input-22-b2cd86e6079b>:1: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples
example using ravel().
rf.fit(xtrain,ytrain)
RandomForestClassifier()
ypred_lr = lr.predict(xtest)
ypred_dt = dt.predict(xtest)
ypred_rf = rf.predict(xtest)
ypred_lr
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
ypred_dt
array([0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1,
0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1])
ypred_rf


eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/adaffb34-90bc-4428-8ce1-5eacd26dfe88/view?projectid=
IBM Watson Studio
Sri Pranesh's Account
Projects / PROJECT1 / Classification (1)
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=10)
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
lr = LogisticRegression()
dt = DecisionTreeClassifier()
rf = RandomForestClassifier()
lr.fit(xtrain,ytrain)
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Ple
ge the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)
LogisticRegression
LogisticRegression()


eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/adaffb34-90bc-4428-8ce1-5eacd26dfe88/view?projectid=
IBM Watson Studio
Projects / PROJECT1 / Classification (1)
confusion_matrix(ytest,ypred_dt)
array([[45, 7],
[1, 27]])
confusion_matrix(ytest,ypred_rf)
array([[47, 5],
[0, 28]])
# Save the model
import pickle
pickle.dump(rf,open('classification_rf.pkl','wb'))

HARDWARE/ SOFTWARE DESIGNING:

Implementation Details: The project will be implemented using the following technologies and tools:

Programming Language: Python

Machine Learning Libraries: Scikit-learn,tensoflow

Natural Language Processing Libraries: NLTK,

Data Storage: Relational or NoSQL database (e.g., MySQL, MongoDB)

Web Development: Flask, Django (for building a user interface if required)

Expected Deliverables: The expected deliverables of the project include:

A trained multilabel classification model capable of accurately categorizing printed media articles into multiple topics.

A scalable and efficient system for media monitoring and analysis.

Documentation outlining the project objectives, methodology, and implementation details.

Source code and scripts for data preprocessing, model training, and system deployment.

Evaluation metrics and results demonstrating the performance of the developed model.

Timeline: The project timeline will be as follows:

Data Collection and Preprocessing: 2 weeks

Feature Extraction and Model Development: 4 weeks

Model Evaluation and Optimization: 2 weeks

System Deployment and Testing: 2 weeks

Documentation and Finalization: 1 week

Note: The timeline is subject to change based on project requirements and constraints.

Source code:

```
Import numpy as np
Import pandas as pd


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='aCITeKqRPg7wYTyKGYjrlFdPCTX807eEtlOJUD0ULeTT',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu-de.cloud-object-storage.appdomain.cloud')

bucket = 'project1-donotdelete-pr-jainrcttz9iwxl'
object_key = 'Social_Network_Ads.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 = pd.read_csv(body)
df.head()


"df = pd.read_csv('/content/Social_Network_Ads.csv')
df.head()"


df.shape
df.info()
df.isnull().sum()
df.isnull().sum()
```

```
df.columns
df.drop(["User ID"],axis=1,inplace=True)
df['Gender'].unique()
df['Gender'] = df['Gender'].replace({'Male':0,'Female':1})
df.info()
x = df.iloc[:,0:3]
x

y = df.iloc[:,3:]
y
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=10)
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
lr = LogisticRegression()
dt = DecisionTreeClassifier()
rf = RandomForestClassifier()
lr.fit(xtrain,ytrain)
dt.fit(xtrain,ytrain)
rf.fit(xtrain,ytrain)
ypred_lr = lr.predict(xtest)
ypred_dt = dt.predict(xtest)
ypred_rf = rf.predict(xtest)
ypred_lr
ypred_dt
ypred_rf
from sklearn.metrics import confusion_matrix
confusion_matrix(ytest,ypred_lr)
confusion_matrix(ytest,ypred_dt)
confusion_matrix(ytest,ypred_rf)
import pickle
pickle.dump(rf,open('classification_rf.pkl','wb'))
!pip install -U ibm-watson-machine-learning
from ibm_watson_machine_learning import APIClient
import json
import numpy as np
```

```
wml_credentials = {
    "apikey":"JDO_vmJfYTUdKabRgIDnatN5RVYNrK_zIsIdEAmNfoEO",
    "url":"https://eu-gb.ml.cloud.ibm.com"
}
wml_client = APIClient(wml_credentials)
wml_client.spaces.list()
wml_client.set.default_space(SPACE_ID)
```

Conclusion:

The Media Monitoring Multilabel Classification project aims to develop a robust and efficient system for categorizing printed media articles into multiple topics. By accurately classifying articles, the system will enable media monitoring and analysis tasks to be performed more effectively. The project will follow a systematic methodology and utilize state-of-the-art machine learning techniques to achieve the desired objectives.