Machine Learning Model Deployment with IBM Cloud Watson Studio

BATCH MEMBERS

732721205015: Gobika . R

732721205017: Gomathi . M

732721205020: Hemamalini . S

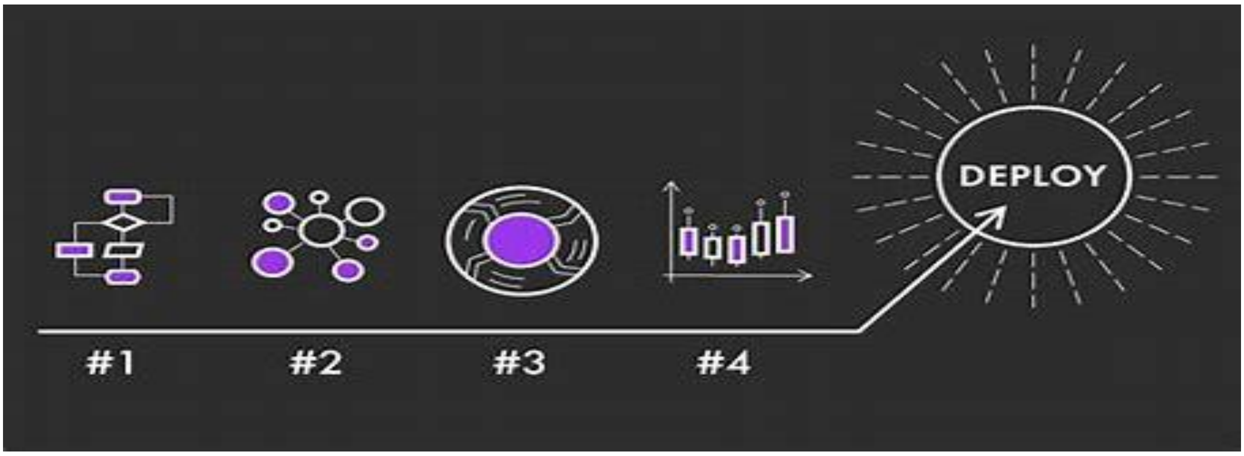
732721205022: Janarthini . S

732721205023: Jayasri . S

Phase 3 Submission document

Project Title: Model Deployment Machine Learning

Topic: Start building the machine learning model using IBM Cloud Watson Studio's tools to import the dataset, preprocess the data, select features, and train machine learning model.



Model Deployment Machine Learning

INTRODUTION:

All analytical assets within Watson Studio are organized into ‘Projects’. Projects are workspaces which can include data assets, Notebooks, Flows and models (among other items). Projects can be personal or shared with a team of collaborators.

1. From the ‘Welcome’ page of Watson Studio click on ‘New project’.

2. Enter a project name, and select your target IBM Cloud Object Storage Instance (*we’ll help you create a storage instance if you don’t have one already*).

3.this graphic illustrates a typical process for a machine learning model.

Define - your project goals

What do you want to find out?
Do you have the data to analyze ?

Prepare - the data

- Refine the data

- Add the data as a project asset or in a data repository.

Choose - a tool

- Pick the tool that matches your data and desired outcome
- Choose between an automated process, a graphical editor, or code your own model

Train - your model

- Train the model with the data you supply
- Let a model building tool choose estimators and optimizers or choose your own

Deploy – your model

- Score the model to generate prediction
- Make your model Available in product
- Retrain as needed

Add data assets to project

Data assets are added to a project to make them available for any of the tools included in Watson Studio. Data assets added to a project are also available for any of the collaborators sharing that project to use. Examples of data assets are files, such as .csv or .json or database tables added through a data connection.

1. Click on the ‘Community’ tab in the top bar. From here you can see content in the Watson Studio Community, which includes data sets.

2. Navigate to the data set ‘**Calls by customers of a Telcom company’**, then click on the add (+) icon on the community card for this data set, select your project and click ‘Add’. Once this completes it will display ‘Added’. This may take up to a minute.

3. From the Community add the data set ‘**Customers of a Telcom including services used**’, and once it displays ‘Added’ click on the ‘View Project’ link.

4. Within the project’s ‘**Assets**’ tab you will see ‘Data assets’ section, which should now have the two added above. Click on the ‘**Calls by customers of a Telcom company.csv**’ name to preview the data. From here you will see the description and name on the right. Click on the name to and rename it to ‘**calls**’ then click ‘Apply’.

5. Return to the project by clicking on the project name in the navigation breadcrumb, then perform similar steps to rename ‘Customers of a Telco including services used.csv’ to ‘**customers**’.

OBJECTIVES

- Introduction to Watson Studio and Watson Machine Learning for Cloud Pak for Data
- Work with analytics projects
- Import data
- Prepare data for modeling with Data Refinery

- Automate building supervised models with AutoAI experiment
- Work with notebooks
- Deploy Watson Machine Learning models

CONTENT

Introduction to Watson Studio and Watson Machine Learning for Cloud Pak for Data

- Describe the IBM Cloud Pak for Data platform and AI
- Describe the four rungs in the ladder to AI
- Describe the personas on the platform
- Describe how to collaborate on the platform
- Describe the CRISP-DM methodology

Work with analytics projects

- Describe analytics projects
- Create analytics projects
- Leverage industry accelerators

Import data

- Identify key concepts in working with data
- Describe correct column types
- Add local files to the project
- Created connections
- Add connected data sets to the project

Prepare data for modeling with Data Refinery

- Identify three tasks in preparing data for modelling
- Describe the capabilities of Data Refinery
- Describe steps, flows, and jobs
- Join data
- Profile data
- Visualize data

Building a flow in Modeler

1. From the navigation bar select ‘Add to project’ then ‘Modeler flow’.

and click ‘Create’.

<u>TAKEN DATA SET:</u>

	rev_Mean	mou_Mean	totmrc_Mean	da_Mean	ovrmou_Mean	ovrrev_Mean	vceovr_Mean	datovr_Mean	roam_Mean	change_mou	...	forgntvl	ethnic	k
0	23.9975	219.25	22.500	0.2475	0.00	0.00	0.00	0.0	0.0	-157.25	...	0.0	N	
1	57.4925	482.75	37.425	0.2475	22.75	9.10	9.10	0.0	0.0	532.25	...	0.0	Z	
2	16.9900	10.25	16.990	0.0000	0.00	0.00	0.00	0.0	0.0	-4.25	...	0.0	N	
3	38.0000	7.50	38.000	0.0000	0.00	0.00	0.00	0.0	0.0	-1.50	...	0.0	U	
4	55.2300	570.50	71.980	0.0000	0.00	0.00	0.00	0.0	0.0	38.50	...	0.0	I	
5	82.2750	1312.25	75.000	1.2375	0.00	0.00	0.00	0.0	0.0	156.75	...	0.0	U	
6	17.1450	0.00	16.990	0.0000	0.00	0.00	0.00	0.0	0.0	0.00	...	0.0	N	
7	38.0525	682.50	52.490	0.2475	0.00	0.00	0.00	0.0	0.0	147.50	...	1.0	S	
8	97.3375	1039.00	50.000	4.9500	419.50	41.95	41.95	0.0	0.0	198.00	...	0.0	F	
9	31.6625	25.50	29.990	0.2475	0.00	0.00	0.00	0.0	0.0	59.50	...	1.0	N	

1.Import Libraries :

Start by importing necessary libraries

Program:

Import pandas as pd

```
Import numpy as np

From sklearn.model_selection import train_test_split

From sklearn.preprocessing import StandardScaler
```

Load the Dataset:

Load your dataset into a Pandas Data Frame. You can typically find the customer calls dataset in csv format , but you can adapt this code to other formats as needed.

Program:

```
df =pd.read_csv("Telecom_customer churn.csv")

pd.read()
```

Exploratory Data Analysis (EDA):

Perform EDA to understand your data better .This includes checking for missing values, exploring the data’s statistics, and visualizing it to identify patterns.

The data set can be pre-processing the below steps.

Program:

#Check for missing values

Print(df.isnull().sum())

#Explore Statistics

Print(df.describe())

Feature Engineering:

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

Split the data:

Split your dataset into training and testing sets. This helps you evaluate your model’s performance later.

Program:

X=df.drop(‘totrev’,axis=1)#Features

y=df[‘totrev’] # Target variable

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

Feature Scaling:

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean = 0 and std=1)is a common choice.

Program:

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.ttranform(X_test)
```

Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step in building any machine learning model. However , it is especially important for **Calls by customers of a Telcom company models,** as model deployment are often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able from the dataset effectively and accurately.

Challenges involved in loading and preprocessing a house price dataset:

There are a number of challenges involved in loading and preprocessing a house customers dataset

- **Handling missing values:** House price datasets often contain missing values, which can be due to a variety of factors, such as human error or incomplete data collection. Common methods for handling missing values include dropping the rows with missing values, imputing the missing values with the mean or median of the feature, or using a more sophisticated method such as multiple imputation.

- **Encoding categorical variables**: House price datasets often contain categorical features, such as the type of house, the neighborhood, and the school district. These features need to be encoded before they can be used by machine learning models. One common way to encode categorical variables is to use one-hot encoding.

- **Scaling the features:**

  It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust outliers. There are a variety of ways to scale the features, such as min-max scaling and standard scaling.

- **Splitting the dataset into training and testing sets:**

Once the data has been pre-processed, we need to split the dataset into training sets. The training set will be used to train the model, and the testing set will be used to evaluated the performance of the model on unseen data. It is important to split the dataset in a way in a way that is representative pf the real world distribution of the data.

How to overcome the challenges of loading and preprocessing a customer calls dataset:

There are a number of things that can be done to overcome the challenges of loading and preprocessing a customer calls dataset, including

- **Use a data preprocessing library:**

  There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.

- **Carefully consider the specific needs of your model:**

  The best way to preprocess the data will depend on the specific machine learning algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to preprocess the data in a way that is compatible with the algorithm.

- **Validate the preprocessed data:**

It is **important** to validate the preprocessed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality. This can be done by inspecting the data visually or by using statistical methods.

1.Loading the dataset:

- ✓ Loading the dataset using Machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.

- ✓ The specific steps involved in loading the dataset will vary depending om the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks

a. **Identify the dataset:**

The first step is to identify the dataset that you want to load. This dataset may be

stored in a local file, in a database, or in a cloud storage service.

b. **Load the dataset:**

Once you have identified the dataset that you need to load it into the machine learning environment. This may involve using a built -in function in the machine learning library, or it may involve writing your own code.

c. **Preprocess the dataset:**

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Here, how to load a dataset using machine learning in python.

Program:

```
import pandas as pd

import numpy as np

import seaborn as sns
```

```
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import
r2_score,mean_absolute_error,mean_squared_error

from sklearn.linear_model import LinearRegression

from sklearn.linear_model import Lasso

from sklearn.ensemble import RamdomForestRegressor

from sklearn.svm import SVR
```

Loading Dataset:

```
df =pd.read_csv("Telecom_customer churn.csv")
```

Data Exploration:

Dataset

2.Preprocessing the dataset:

- Data preprocessing is the process of cleaning, transforming, and integrating data in order to make it ready for analysis.

- This may involve removing errors and inconsistencies, handling missing values, transforming the data into a consistent format, and scaling the data to a suitable range.

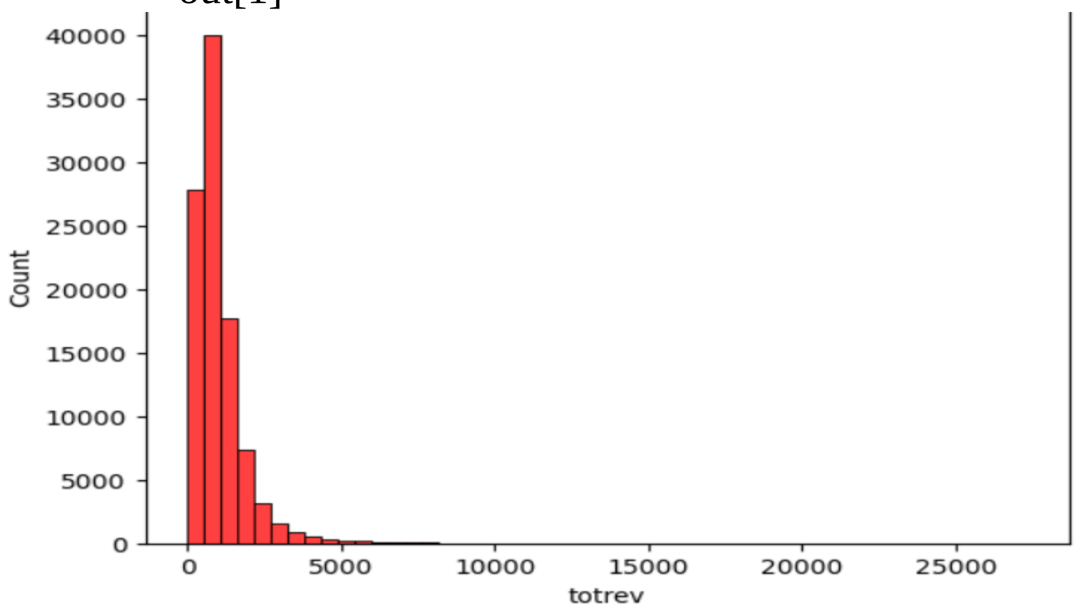
Visualisation and pre-processing of Data:

In[1]:

sns.histplot(df, x='totrev', bins=50, color='r')

Out[1]:
<Axes:xlable='totrev', ylabel='Count'>

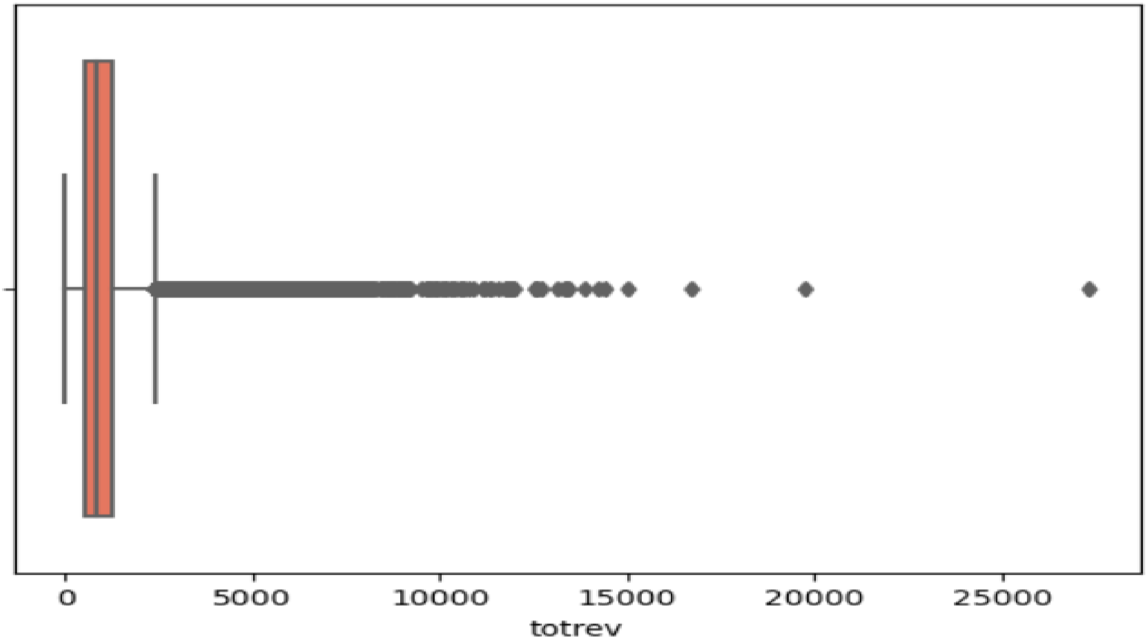
out[1]



In[2]:

```
sns.boxplot(df, x='totrev', palette='Reds')
```

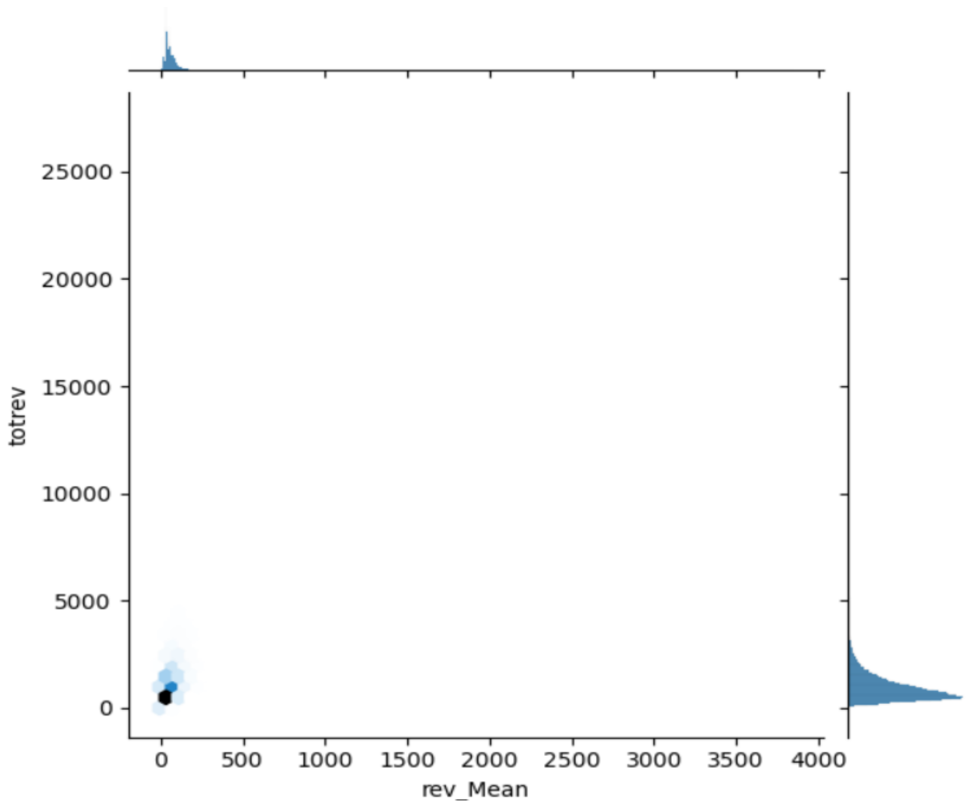
Out[2]:
<Axes: xlabel='totrev'>



In[3]:

```
sns.jointplot(df, x='rev_Mean', y='totrev', kind='hex')
```

Out[3]:
<seaborn.axisgrid.JointGrid at 0x184c3a53e50>



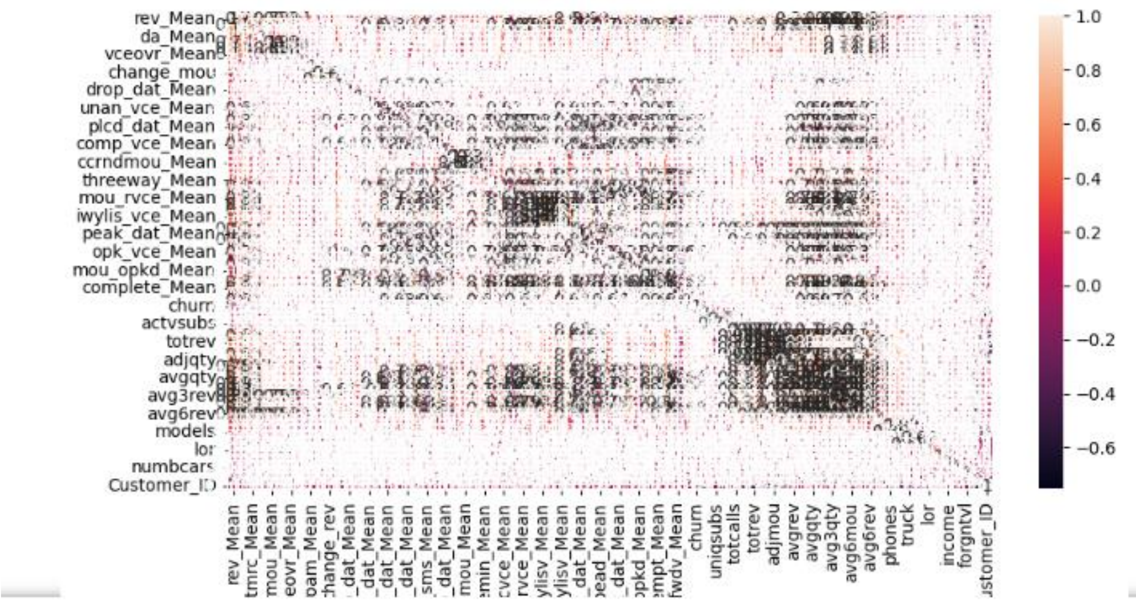
In[4]:

```
plt.figure(figsize=(10,5))

sns.heatmap(df.corr(numeric_only = True), annot= True)
```

Out[4]:

<Axes: >



Some common data preprocessing tasks include:

- **cleaning:** This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missing values.

- Data transformation: This involves converting the data into a format that is suitable for the analysis task. For example, this may involve converting categorical data to numerical data, or scaling the data to a suitable range.

- **Feature engineering:** This involves creating new features from the existing data. For example, this may involve creating features that represent interactions between variables, or features that represent summary statistics of the data.

- **Data integration:** This involves combining data from multiple sources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

Data preprocessing is an essential step in many data science projects. By carefully preprocessing the data, data scientists can improve the accuracy and reliability of their results.

Conclusion:

- In the quest to build a house price prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset. We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.

- Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) is essential for informed decision-making.

- Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensure that it aligns with the requirements of machine learning algorithms.