IBM Phase 5

MACHINE LEARNING MODEL

DEPLOYMENT USING IBM WATSON

STUDIO

PROJECT TITLE: LAPTOP PRICE PREDICTOR

**Phase 2: Innovation**

**Concept:**

The Laptop Price Predictor application is designed to help users estimate the price of laptop devices based on their features. The machine learning model is trained on a dataset of mobile features and corresponding prices, providing accurate predictions for a wide range of devices. By leveraging IBM

Watson services for the frontend, the application offers a user-friendly interface and seamless deployment.

**Features:**

Search and Filtering: Users can search for laptops based on

specific criteria such as brand, specifications, price range, and

more. Advanced filtering options help users narrow down their

choices.

Historical Price Data: Users may access historical price data

for laptops, allowing them to see price trends over time and

make informed decisions about when to buy.

Mobile and Web Accessibility: Laptop price predictors may be

accessible through both web browsers and mobile apps,

ensuring users can access the tool on their preferred devices.

**Benefits:**

Budget Planning: Shoppers can plan their budgets more effectively

when they have a reliable price predictor. They can set realistic

expectations for the type of laptop they can afford.

Time Savings: Instead of manually researching and comparing

laptop prices across different retailers and brands, users can quickly

obtain price estimates, saving them time and effort.

Price Alerts: Users can set price alerts to be notified when a laptop

they are interested in reaches a certain price point, allowing them to

make a purchase when the price is right.

Financial Planning: For organizations or individuals purchasing

laptops in bulk, a price predictor can help with financial planning and

Budgeting.

The laptop Price Predictor application is designed to help users

estimate the price of laptop devices based on their features.

The machine learning model is trained on a dataset of laptop

features and corresponding prices, providing accurate

predictions for a wide range of devices. By leveraging IBM

Watson services and Streamlit for the frontend, the application

offers a user-friendly interface and seamless deployment.

Backup and Replication

In this subphase , we need to configure automated, regular backups

of your virtual server instances and data using IBM Cloud services

like IBM Cloud Object Storage and I Implement data replication to

another geographic region using IBM Cloud&#39;s Cross Region

Resiliency or similar services.

Network and Connectivity

Network and connectivity is an important role in this phase.

Establish a secure and reliable network connection between your

primary and failover data centers using VPN, Direct Link, or similar

technologies. Also ,test network configurations to ensure that

applications and services will function seamlessly during a failover

event.

Features

 Intuitive web-based user interface powered by

Streamlit

 Machine learning model deployment via IBM

Watson services

 API integration for easy access to the

deployed model

 Let&#39;s break down the project into more detail starting from setting

up the IBM Cloud account and provisioning Watson studio:

**Setting Up an IBM Cloud Account :**

1. Once you are on the account creation page, follow

the below instructions to create an IBM cloud trial

account.

2. Enter your Email address [preferably use Gmail ID

or Yahoo ID] and a strong Password, as per criteria

and then click the Next button.

3. Check your email and copy and paste Verification

code.

4. Please ensure you click on the &quot;Open Tool&quot; button

5. Open tool to get a feature code and activate the trial

account .

6. Create a project in IBM Watson Studio Dashboard and assign a

Cloud object Storage service to manage datasets

7. Note: Cloud Object Storage is a storage service in IBM

Cloud. We use this service to manage our datasets for training

the ML Model and store required files.

8.

9. Add a jupyter notebook instance in your project to Develop and

Deploy Machine Learning Model.

10. Note: You can either create a blank notebook or import from

existing file or URL.

Build a Machine Learning model using jupyter notebook instance.

Installation &amp; Setup

To set up the laptop Price Predictor application on your

local machine, follow these steps:

1. Clone the repository using

2. Navigate to the project directory using

3. Create a virtual environment and activate it

4. Install the required dependencies using

pip install -r requirements.txt

Let’s start Building :

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

next step we are going to upload and insert a laptop dataset for

training our Machine Learning model as pandas dataframe.

Once the dataset is imported you can proceed further with pre-

processing steps and building the model .

#Check Missing Values

dataset.isnull().any()

#Spilt Dependent and Independent Variables

X = dataset.iloc[:, [2, 3]].values

y = dataset.iloc[:, 4].values

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size= 0.25, random\_state = 0)

# Fitting Decision Tree Classification to the Training set

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(criterion = &#39;entropy&#39;,

random\_state = 0)

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

# Predicting the Test set results

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

#Finding the accuracy score

from sklearn.metrics import accuracy\_score

print(&quot;Accuracy Score: &quot;,accuracy\_score(y\_test,y\_pred)\*100,&quot;%&quot;)

create Machine Learning service in AI Category.

Import WatsonMachineLearningAPIClient library. Watson studio

uses Watson Machine Learning service credentials to access WML service.

from watson\_machine\_learning\_client import

WatsonMachineLearningAPIClient

wml\_credentials={

&quot;url&quot;: &quot;xxxxxxxxxxxxxxxxxx&quot;,

&quot;apikey&quot;: &quot;xxxxxxxxxxxxxxxxxxxxxxx&quot;,

&quot;username&quot;: &quot;xxxxxxxxxxxxxxxxxxxx&quot;,

&quot;password&quot;: &quot;xxxxxxxxxxxxxxxxxxxxxxx&quot;,

&quot;instance\_id&quot;: &quot;xxxxxxxxxxxxxxxxxxxxxx&quot;

}

client = WatsonMachineLearningAPIClient(wml\_credentials)

we have to specify our machine learning model properties and store the

model in WML repository.

#Specify the Properties

model\_props = {client.repository.ModelMetaNames.AUTHOR\_NAME: &quot;Abhi&quot;,

client.repository.ModelMetaNames.AUTHOR\_EMAIL: &quot;&quot;,

client.repository.ModelMetaNames.NAME: &quot;MyModel&quot;}

#Store the Machine Learning Model

model\_artifact=client.repository.store\_model(classifier,

meta\_props=model\_props)

Now ,we can find the list of saved models

client.repository.list()

In this phase , Now we are ready to deploy our machine learning

model as a Web service.

To promote the model to deployment you must specify a

deployment space. If no space is created choose the New space

+ option to create one. This action will associate the model with

the space.

Navigate to the space using the hamburger menu (☰) on the

top right and choose to View all spaces.

Choose the deploy the model by clicking the rocket ship icon.

Your new deployment will appear. Click on the API

reference tab and save the Endpoint. We&#39;ll be using this in our

application.

You can deploy this application as a Cloud Foundry application

to IBM Cloud by simply clicking the button below. This option

will create a deployment pipeline, complete with a hosted Git

lab project and devlop toolchain.

You may be prompted for an IBM Cloud API Key during this

process. Use the Create (+) button to auto-fill this field and the

others. Click on the Deploy button to deploy the application.

Before using the application go to the Runtime section of the

application and in the Environment variables tab add in

your API\_KEY and DEPLOYMENT\_URL values.

Once updated your application will restart and you can visit the

application by clicking on Visit App URL.

The app is fairly self-explantory, simply fill in the data you want

to score and click on the Classify button to test how those

figures would score against our model. The model predicts that

the price of the laptops based on their specifications.

So , what is an application here ??

 In the context of the project we&#39;re discussing, an &quot;application&quot;

refers to the software or program that is hosted on your virtual

server within IBM Cloud. It&#39;s the software that provides specific

functionality, whether it&#39;s a website, web service, database, or

any other type of program.

Here are a few examples to illustrate what an &quot;application&quot; could be:

1. Website: Your application could be a simple website that

provides information about a topic, a blog, or an e-commerce

site.

2. Web Service: It could be a RESTful API or a web service

that provides functionality to other software applications.

3. Database: The application could be a database server that

stores and manages data, such as customer records, product

information, or financial transactions.

4. Content Management System (CMS): It might be a CMS

like WordPress that powers a blog or content-driven website.

5. Custom Software: The application could be custom software

that you or your team have developed for a specific purpose,

such as inventory management, order processing, or customer

relationship management.

This innovation leverages IBM's cloud capabilities ,laptop price predictor and

machine learning technology will produce the price results of the laptops that

the laptops may have low price for a specific period of time then in this case ,

The laptop price predictor gives the price alert to the users.