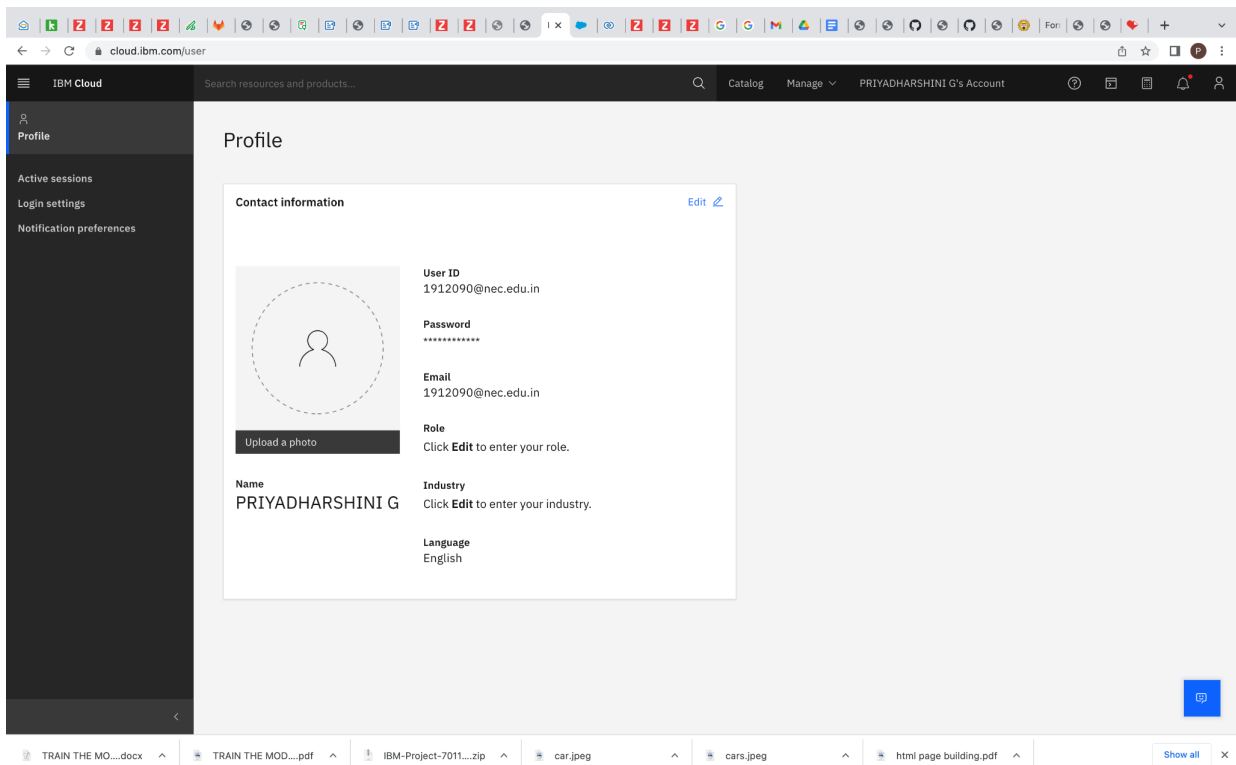
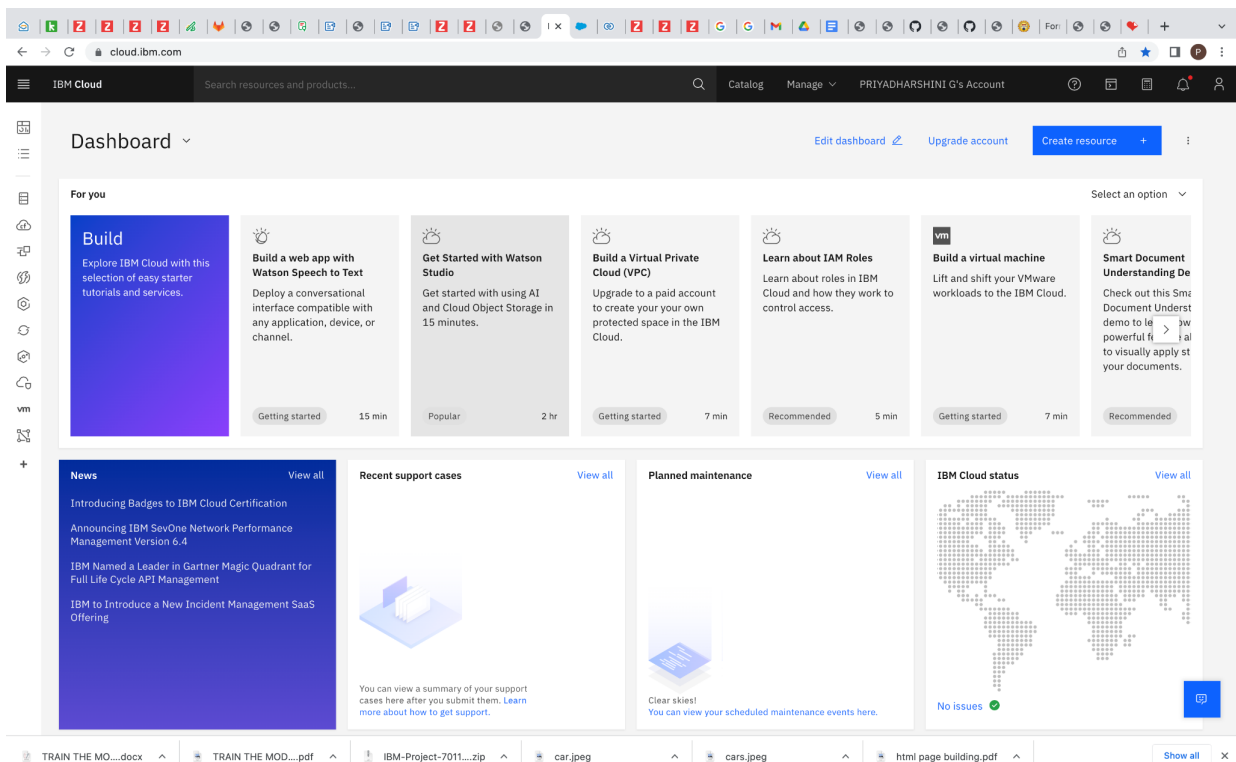


TRAIN THE MODEL ON IBM

1. REGISTER FOR IBM CLOUD



2. TRAIN THE ML MODEL ON IBM



The screenshot shows the IBM Watson Studio interface. The top navigation bar includes the IBM logo, a search bar, and user account information. The main content area displays the 'Data asset' page for 'Pre-Process the Data.ipynb'. A summary table provides details about the asset, and a data preview section shows the first row of the notebook's output.

Name	Pre-Process the Data.ipynb
ID	32544ad3-f6c6-4453-a569-efd928b66d2e
Date created	Nov 11, 2022, 3:52 PM
Size	48.8 KB
Data source type	IBM Cloud Object Storage
MIME type	application/octet-stream

Data preview (Previewing the first row)

cells	metadata	nbformat	nbformat_minor
[{"cell_type": "markdown", "id": "T06305da", "metadata": {}, "source": ["# Import Libraries"]}, {"cell_type": "code", "execution_count": 1, "id": "7b891062", "metadata": {}, "outputs": [{"source": ["import pandas as pd", "import numpy as np", "import matplotlib as plt", "from sklearn.preprocessing import LabelEncoder"], "import pickle"}], "cell_type": "markdown", "id": "414f742a", "metadata": {}, "source": ["Import Dataset"]}]			

3. INTEGRATE FLASK WITH SCORING END POINT

The screenshot shows the IBM Watson Studio interface with a Jupyter notebook open. The notebook contains Python code for integrating Flask with a scoring endpoint. The code defines a Flask app, sets up a route, and uses a LabelEncoder to preprocess data before making a prediction.

```

In [28]: labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

In [29]: mapper = {}
for i in labels:
    mapper[i] = LabelEncoder()
    mapper[i].fit(new_df[i])
    tr = mapper[i].transform(new_df[i])
    np.save(str('classes'+i+'.npy'), mapper[i].classes_)
    print(i, "-", mapper[i])
    new_df.loc[:, i + '_labels'] = pd.Series(tr, index=new_df.index)

gearbox : LabelEncoder()
notRepairedDamage : LabelEncoder()
model : LabelEncoder()
brand : LabelEncoder()
fuelType : LabelEncoder()
vehicleType : LabelEncoder()

In [31]: labeled=new_df[ ['price'
                        , 'yearOfRegistration'
                        , 'powerPS'
                        , 'kilometer'
                        , 'monthOfRegistration'
                        ]
          + [x+"_labels" for x in labels]]

In [32]: print(labeled.columns)

Index(['price', 'yearOfRegistration', 'powerPS', 'kilometer',
      'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_labels',
      'model_labels', 'brand_labels', 'fuelType_labels',
      'vehicleType_labels'],
      dtype='object')

```