

# TRAINING THE ML MODEL ON IBM CLOUD

IBM Watson Studio

Search in your workspaces

Buy

?

ARUN A's Account

Dallas

AA

Projects / Car Resale Value Prediction IBM / Resale\_Value\_Model

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

In [27]:

```
import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle
```

In [28]:

```
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='2Xcapalh0rE5A6fyhLSmRKvIyegsLMgW_Xun2HAMpZHg',
                              ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
                              config=Config(signature_version='oauth'),
                              endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'carresalevaluepredictionibm-donotdelete-pr-cdchbq0a85f2oq'
object_key = 'autos_preprocessed.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()
```

Out[28]:

Unnamed: 0	price	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage
0	1	18300	coupe	2011	manual	190 not-declared	125000	5	diesel	audi	Yes

Trusted | Python 3.9  ↗

Run

```
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)
labeled = new_df[ ['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration'] + [x+'_'+labels" for x in labels]]

gearbox ; LabelEncoder()
notRepairedDamage ; LabelEncoder()
brand ; LabelEncoder()
fuelType ; LabelEncoder()
vehicleType ; LabelEncoder()
```

```
In [35]: print(labeled.columns)

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

```
In [36]: Y = labeled.iloc[:,0].values
         X = labeled.iloc[:,1:].values
```

```
In [37]: Y = Y.reshape(-1,1)
from sklearn.model_selection import cross_val_score , train_test_split
X train,X test,Y train,Y test = train test split(X,Y,test_size=0.3,random state=3)
```

```
In [38]: from sklearn.ensemble import RandomForestRegressor
         from sklearn.metrics import r2_score
         regressor = RandomForestRegressor(n_estimators = 1000,max_depth=10,random_state=34)
```

```
In [39]: regressor.fit(X_train,np.ravel(Y_train,order='C'))
```

```
Out[39]: RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)
```

```
In [40]: y_pred = regressor.predict(X_test)
print(r2_score(Y_test,y_pred))
```

0.8282067869983389

Note: Only first 50 records were displayed. To display more use 'limit' parameter.

```
Out[84]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

```
Out[89]: '36dcfae5-29ee-4226-89a2-9e5a9a9ab2af'
```

```
Out[93]: array([[ 2002,   116, 150000,    10,     1,     0,    10,     1,
                   1])
```

```
Out[95]: array([2596.34923754])
```

# DEPLOYMENT OF THE MODEL IN IBM CLOUD

Search in your workspaces

Buy

ARUN A's Account

Dallas

AA

Press **F11** to exit full screen

Welcome, Arun!

**Take a tutorial**  
Step through implementing a Data fabric use case in a sample project.  
→

**Work with data**  
Create a project for your team to prepare data, find insights, or build models.  
→

**Learn what's new**  
Stay current with new features, enhancements, and other changes.  
→

**Quick start**

Create data pipelines with DataStage

Build customer profiles with IBM Match 360 with Watson

Catalog and govern data with Watson Knowledge Catalog

Build and manage ML models with Watson Studio

Query data anywhere with Watson Query

**Projects** +

Car Resale Value Prediction IBM

Yesterday at 01:46 PM

**New in gallery**

**SAMPLE PROJECT**

**AI governance**

Tutorials in this project: Build and deploy a machine learning model to predict which applicants qualify for mortgages; Test and validate the model for fairness, accuracy, trustworthiness, and readiness for production.

**Notifications**

**Online deployment ready**  
The online deployment [CRVP\\_Deployment](#) in space [CRVP\\_Model](#) is ready to accept  
Yesterday at 04:59 PM

**Deployments** ⓘ +

CRVP\_Model

Yesterday at 04:56 PM

**Feedback**

**What's new**

New Runtime 2022 release for Python 3.10

# CRVP\_Model

Overview

Assets

Deployments

Jobs

Manage

🔍 Search

Name	Type	Status	Asset	Last modified	
CRVP_Deployment	Online	✔️ Deployed	Car_Resale_Value_Prediction_IBM	10 hours ago ARUN A (You)	⋮

CRVP\_Deployment

Deployed

Online

API reference

Test

Direct link

Endpoint

https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9aaee049-3c55-47c0-ae13-206286158edf/predictions?version=2022-11-18

Bearer <token>

IAM

Code snippets

- cURL
- Java
- JavaScript
- Python
- Scala

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values": [array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9aaee049-3c55-47c0-ae13-206286158edf/predictions?version=2022-11-18', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
```

# TESTING THE DEPLOYMENT

The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `main.py` with the following code:

```
1 import requests
2 import json
3 API_KEY = "HTh0LLSFbzW1vaPHZ2JAZGYK_tAqS5FGQcHd4e6AkQF"
4 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
5 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
6 mltoken = token_response.json()["access_token"]
7
8 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
9 # NOTE: manually define and pass the array(s) of values to be scored in the next line
10 payload_scoring = {"input_data": [{"field": [{"price", 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS', 'model', 'kilometer', 'monthOfRegistration',
11
12 response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9aeee049-3c55-47c0-ae13-206286158edf/predictions?version=2022-11-18',
13 headers={'Authorization': 'Bearer ' + mltoken})
14 print("Scoring response")
15 print(response_scoring.json())
```

The Run tool window at the bottom shows the execution of the script, outputting the following JSON response:

```
Scoring response
{'predictions': [{'fields': ['prediction'], 'values': [[2596.3492375397814]]}]}
```

Below the output, it states "Process finished with exit code 0". The status bar at the bottom indicates the file encoding is UTF-8, the line length is 6:1, and the Python version is 3.9.