

Team ID	PNT2022TMID18719
Project Name	Car Resale value Prediction

Train The ML Model On IBM

The screenshot shows the IBM Watson Studio dashboard. At the top, there's a navigation bar with 'IBM Watson Studio', a search bar, and user account information. The main area features a 'Welcome, Krithika sree!' message and three primary action cards: 'Take a tutorial', 'Work with data', and 'Learn what's new'. Below these, there's a 'Quick start' section with links to create data pipelines, build customer profiles, catalog and govern data, build and manage ML models, and query data anywhere. The 'Projects' section shows the 'Car Resale value Prediction' project. The 'Notifications' section shows 'No notifications'. The 'Deployments' section shows a deployment named 'CAR'. The 'New in gallery' section highlights 'AI governance' as a sample project.

The screenshot shows the 'Assets' page for the 'Car Resale value Prediction' project. The page has tabs for 'Overview', 'Assets', 'Jobs', and 'Manage'. The 'Assets' tab is active, showing a list of 3 assets. On the left, there's a sidebar with 'Asset types' including 'Data' (2 items) and 'Notebooks' (1 item). The main area displays a table of 'All assets' with columns for 'Name' and 'Last modified'. The assets listed are: 'IBM Train the model' (Notebook, 5 hours ago), 'resale_model.sav' (application/octet-stream, 5 hours ago), and 'autos.csv' (CSV, 6 hours ago). On the right, there's a 'Data in this project' section with a drop zone for uploading files.




Name	Last modified
IBM Train the model Notebook	5 hours ago Modified by you
resale_model.sav application/octet-stream	5 hours ago Modified by you
autos.csv CSV	6 hours ago Modified by you

CAR

Overview Assets **Deployments** Jobs Manage

▼

Q Search

Name	Type	Status	Asset	Last modified	↓
 Model_IBM	Online	 Deployed	Car_Resale_value_Prediction	21 seconds ago Krithika sree T (You)	

Items per page: 20 1–1 of 1 items

1 of 1 pages

```
In [1]: import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle
print("IMPORTED REQUIRED LIBRARIES")

IMPORTED REQUIRED LIBRARIES

In [2]: import os, types
import pandas as pd
from boto3.client import Config
import ibm_boto3
import io

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='rH67HclbfmQwWdp-tCvqTblyYtLCRc3xIVcbQzfof',
                              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 = 'carresalevalueprediction-donotdelete-pr-51z8jksdyhlus'
object_key = 'autos.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)

df = pd.read_csv(io.BytesIO(body["Body"].read()), header=0 , sep=',', encoding='Latin1', low_memory=False)
df.head()
```

```
Out[2]:
```

	dateCrawled	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage	dateCre
0	2016-03-24 11:52:17	Golf_3_1.6	privat	Angebot	480	test	NaN	1993	manuell	0	golf	150000	0	benzin	volkswagen	NaN	2016-C 00f
1	2016-03-24 10:58:45	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuell	190	NaN	125000	5	diesel	audi	ja	2016-C 00f
2	2016-03-14 12:52:21	Jeep_Grand_Cherokee_Overland	privat	Angebot	9800	test	suv	2004	automatik	163	grand	125000	8	diesel	jeep	NaN	2016-C 00f
3	2016-03-17 16:54:04	GOLF_4_1.4_3TURER	privat	Angebot	1500	test	kleinwagen	2001	manuell	75	golf	150000	6	benzin	volkswagen	nein	2016-C 00f
4	2016-03-31 17:25:20	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	2008	manuell	69	fabia	90000	7	diesel	skoda	nein	2016-C 00f

```
In [3]: print(df.seller.value_counts())
df[df.seller != 'gewerblich']
df=df.drop('seller',axis=1)

print(df.offerType.value_counts())
df[df.offerType != 'Gesuch']
df=df.drop('offerType',axis=1)

privat      371525
gewerblich      3
Name: seller, dtype: int64
Angebot      371516
Gesuch        12
Name: offerType, dtype: int64
```

```
IBM Watson Studio Search in your workspaces Buy Krithika sree T's Account Dallas KT

Projects / Car Resale value Prediction / IBM Train the model

In [4]: print(df.shape)
df=df[(df.powerPS>50) & (df.powerPS<900)]
print(df.shape)
df=df[(df.yearOfRegistration>1950)&(df.yearOfRegistration<2022)]
print(df.shape)
(371528, 18)
(319709, 18)
(319641, 18)

In [5]: df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCode','dateCreated'], axis='columns',inplace=True)

In [6]: new_df=df.copy()
new_df=new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType','notRepairedDamage'])

In [7]: new_df.gearbox.replace({'manuel','automatic'},{'manual','automatic'},inplace=True)
new_df.fuelType.replace({'benzin','andere','elektro'},{'petrol','others','electric'},inplace=True)
new_df.vehicleType.replace({'kleinwagen','cabrio','kombi','andere'},{'small car','convertible','combination','others'},inplace=True)
new_df.notRepairedDamage.replace({'ja','nein'},{'Yes','No'},inplace=True)

In [8]: new_df=new_df[(new_df.price>100)&(new_df.price<150000)]

new_df['notRepairedDamage'].fillna(value='not-declared',inplace=True)
new_df['fuelType'].fillna(value='not-declared',inplace=True)
new_df['gearbox'].fillna(value='not-declared',inplace=True)
new_df['vehicleType'].fillna(value='not-declared',inplace=True)
new_df['model'].fillna(value='not-declared',inplace=True)

In [9]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"GUHVuufH0YvdaXTEnaSHFze4ydlPqFuqypAmlH21C8"
}
client =APIClient(wml_credentials)
```

```
IBM Watson Studio Search in your workspaces Buy Krithika sree T's Account Dallas KT

Projects / Car Resale value Prediction / IBM Train the model

In [10]: def guide_from_space_name(client, space_name):
space = client.spaces.get_details()
# print(space)
return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])

In [11]: space_uid=guide_from_space_name(client,'CAR')
print("Space UID"+ space_uid)
Space UID4c51c2eb-311b-4e62-95e6-1bb3532301d

In [12]: client.set.default_space(space_uid)
client.software_specifications.list()

-----
NAME ASSET_ID TYPE
default_py3.6 0862b3b9-8b7d-44a0-a9b9-46c416adc8d9 base
kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base
pytorch-onnx_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base
scikit-learn_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base
spark-mllib_3.0-scala_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base
pytorch-onnx_rt22.1-py3.9 0b848d44-e681-5599-ba41-b5f6fccc6471 base
ai-function_0.1-py3.6 0cd0ff1e-5376-4f46-92dd-da3b69a990da base
shiny-r3.6 0e6e794f-875e-4f24-8ae9-02dcc2148306 base
tensorflow_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base
pytorch_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base
tensorflow_1.15-py3.6-ddl 111e41b3-de2d-5422-ad46-bf776828c4b7 base
autoai-kb_rt22.2-py3.10 125b6d9a-5b1f-5e8d-972a-b251688ccf40 base
runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base
scikit-learn_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base
default_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base
pytorch-onnx_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base
kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base
pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d8880bde37f base
tensorflow_2.1-py3.6 1eb25b84-d6ed-5d0e-bb45-3f8df1166566 base
spark-mllib_3.2 20047f72-0a98-58cf-9ff5-a77b012eb8f5 base
tensorflow_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base
runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base
do_py3.8 295addb5-9ef9-547e-9bf4-92ae356e720 base
```

```
IBM Watson Studio Search in your workspaces Buy Krithika sree T's Account Dallas KT

Projects / Car Resale value Prediction / IBM Train the model

autoai-kb_3.1-py3.7 632d4b22-10aa-5180-88f0-f52df6444d7 base
-----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.

In [13]: software_spec_uid = client.software_specifications.get_uid_by_name("runtime-22.1-py3.9")
software_spec_uid

Out[13]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'

In [14]: print(new_df)

   price  vehicleType  yearOfRegistration  gearbox  powerPS \
1    18300         coupe                2011    manual      190
2     9800         suv                  2004    automatic    163
3    1500         small car                2001    manual       75
4    3600         small car                2008    manual       69
5     650         limousine               1995    manual      102
...      ...      ...
371520  3200         limousine               2004    manual      225
371524  1199         convertible            2000    automatic    101
371525  9200         bus                  1996    manual      102
371526  3400         combination            2002    manual     100
371527  28990         limousine            2013    manual     320

   model  kilometer  monthOfRegistration  fuelType  brand \
1  not-declared    125000                  5    diesel    audi
2      grand      125000                  8    diesel    jeep
3      golf    150000                  6    petrol  volkswagen
4      fabia    90000                  7    diesel    skoda
5        3er    150000                 10    petrol    bmw
...      ...      ...
371520      leon    150000                  5    petrol    seat
371524      fortwo    125000                  3    petrol    smart
371525      transporter    150000                  3    diesel  volkswagen
371526      golf    150000                  6    diesel  volkswagen
371527      m_reihe    50000                  8    petrol    bmw

   notRepairedDamage
1                  Yes
```

```
IBM Watson Studio Search in your workspaces Buy Krithika sree T's Account Dallas KT

Projects / Car Resale value Prediction / IBM Train the model

notRepairedDamage
1 Yes
2 not-declared
3 No
4 No
5 Yes
...
371520 Yes
371524 No
371525 No
371526 not-declared
371527 No

[288028 rows x 11 columns]

In [15]: labels=['gearbox','notRepairedDamage','model','brand','fuelType','vehicleType']

In [16]: mappers={}
for i in labels:
    mapper[i]=LabelEncoder()
    mapper[i].fit(new_df[i])
    trsmapper[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 [17]: labeled = new_df[['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']+xs_labels] for x in labels]]
print(labeled.columns)
Y=labeled.iloc[:,0].values
X=labeled.iloc[:,1:].values
```

```
IBM Watson Studio Search in your workspaces Buy Krithika sree T's Account Dallas KT

Projects / Car Resale value Prediction / IBM Train the model

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

In [18]: YeY.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)
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
regressor = RandomForestRegressor(n_estimators = 1000,max_depth = 10,random_state = 34)

In [19]: regressor.fit(X_train, np.ravel(Y_train,order='C'))
y_pred = regressor.predict(X_test)
print(r2_score(Y_test,y_pred))
filename='resale_model.sav'
pickle.dump(regressor,open(filename,'wb'))
0.8321579621885576

In [20]: model_details = client.repository.store_model(model=regressor,meta_props={
client.repository.ModelMetaNames.NAME: "resale_model",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid,
client.repository.ModelMetaNames.TYPE: "scikit-learn-1.0"
})
model_id = client.repository.get_model_id(model_details)

In [21]: model_id

Out[21]: '2ae62ba4-6e89-41ab-8335-2a0a582f9977'

In [22]: client.repository.download(model_id,'model.tar.gb')

Successfully saved model content to file: 'model.tar.gb'

Out[22]: '/home/user/work/model.tar.gb'
```

```
dataplatform.cloud.ibm.com/analytics/notebooks/v2/b9a4c97f-22c8-4b04-8cf1-6fe24e4275c2/view?projectid=95f3fbee-8ce7-4ae5-ad9f-779e4341e796&context=cpdaas

IBM Watson Studio Search in your workspaces Buy Krithika sree T's Account Dallas KT

Projects / Car Resale value Prediction / IBM Train the model

In [23]: X_train[0]
regressor.predict([[2012.0, 179.0, '1500000', 12.0, 0, 0, 30, 1, 1, 4]])

Out[23]: array([19502.21853325])

Deployment

In [24]: pip install ibm_watson_machine_learning

Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm_watson_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (3.3)
Requirement already satisfied: charset-normalizer==2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)
Requirement already satisfied: pyparsing<3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

```
In [25]: from ibm_watson_machine_learning import APIClient
import json
import numpy as np

In [26]: wml_credentials={"url":"https://us-south.ml.cloud.ibm.com",
                        "apikey":"LSZaHPYb7-vuq7rcgHmMYnIa6yVzIkfIELFdN7pZFF"}

In [27]: client = APIClient(wml_credentials)

In [29]: client.spaces.list()

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
-----
ID          NAME      CREATED
4c51c2eb-311b-4e62-95e6-1bb35332301d  CAR      2022-11-19T07:09:17.483Z
-----

In [31]: space_uid="4c51c2eb-311b-4e62-95e6-1bb35332301d"

In [32]: client.set_default_space(space_uid)

Out[32]: 'SUCCESS'

In [33]: client.software_specifications.list()

-----
NAME          ASSET_ID      TYPE
default_py3.6  0062b8c9-8b7d-44a0-a9b9-46c416adcb09  base
kernel-spark3.2-scala2.12  020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt  069ea134-3346-5748-b513-4912ba15d788  base
scikit-learn_0.20-py3.6  09c5a1d0-9c1e-4473-a344-ab7b665ff687  base
spark-mllib_3.0-scala_2.12  09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9  0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6  0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6  0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod  1099590a-307d-563a-9862-4eb7064b3f22  base
pytorch_1.1-py3.6  10ac12d6-6b30-4ccd-8392-3e922c896a52  base
-----
```

```
runtime-22.2-py3.10-nc  2c232309b01e-72e0-972d-82e7c00a2c0e  base
spark-mllib_2.4-scala_2.11  55a70f99-7320-4be5-9fb9-9e4b5a443af5  base
spark-mllib_3.0  5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1  5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8  5d3232bf-c86b-5d64-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-nc  5e8cd8ff-e84a-5a6a-bbaa-2d4af9864dab  base
autoai-kb_3.1-py3.7  632d4b22-10aa-5180-88f0-f52dfb6444d7  base
-----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

Save and deploy the Model

```
In [34]: import sklearn
sklearn.__version__

Out[34]: '1.0.2'

In [36]: MODEL_NAME = 'Car_Resale_value_Prediction'
DEPLOYMENT_NAME = 'Model_IBM'
DEMO_MODEL = regressor

In [38]: software_spec_uid=client.software_specifications.get_id_by_name('runtime-22.1-py3.9')

In [39]: model_props={
    client.repository.ModelMetaNames.NAME: MODEL_NAME,
    client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}

In [43]: model_details = client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=X_train,
    training_targets=Y_train
)
```

```
In [44]: model_details

Out[44]: {'entity': {'hybrid_pipeline_software_specs': [],
                    'label_column': 'l0',
                    'schemas': {'input': [{'fields': [{'name': 'f0', 'type': 'int'},
                                                         {'name': 'f1', 'type': 'int'},
                                                         {'name': 'f2', 'type': 'int'},
                                                         {'name': 'f3', 'type': 'int'},
                                                         {'name': 'f4', 'type': 'int'},
                                                         {'name': 'f5', 'type': 'int'},
                                                         {'name': 'f6', 'type': 'int'},
                                                         {'name': 'f7', 'type': 'int'},
                                                         {'name': 'f8', 'type': 'int'},
                                                         {'name': 'f9', 'type': 'int'}],
                                'id': '1',
                                'type': 'struct'}],
                    'output': []},
                    'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
                                     'name': 'runtime-22.1-py3.9',
                                     'type': 'scikit-learn_1.0'},
                    'metadata': {'created_at': '2022-11-19T14:44:39.790Z',
                                'id': '36d4a98e-fb29-4f76-bbfc-ad48d74fcbe2',
                                'modified_at': '2022-11-19T14:45:18.833Z',
                                'name': 'Car_Resale_value_Prediction',
                                'owner': 'IBMId-668000FXE7',
                                'resource_key': 'bbb25625-38fa-4e7b-ae4d-cc5b7aaf189f',
                                'space_id': '4c51c2eb-311b-4e62-95e6-1bb35332301d',
                                'system': {'warnings': []}}

In [45]: model_id = client.repository.get_model_id(model_details)

In [46]: model_id

Out[46]: '36d4a98e-fb29-4f76-bbfc-ad48d74fcbe2'
```

IBM Watson Studio

Search in your workspaces

Buy

Krithika sree T's Account

Dallas

KT

Projects / Car Resale value Prediction / IBM Train the model

In [48]:

deployment_props = {
 client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
 client.deployments.ConfigurationMetaNames.ONLINE: {}
}

In [50]:

deployment = client.deployments.create(
 artifact_uid = model_id,
 meta_props = deployment_props
)

#####

Synchronous deployment creation for uid: '36d4a98e-fb29-4f76-bbfc-ad48d74fc0e2' started

#####

initializing

Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready

Successfully finished deployment creation, deployment_uid='b5e57636-f3a7-4791-a1b5-4c2080388dc5'
