TELECOM CUSTOMER CHURN PREDICTION USING IBM WATSON AUTO AI

Final Report

Submitted By: **Sidhant Jain**

Internship Title:RSIP Carrer Basic ML 158

Project ID:SPS_PRO_285

Table of Contents

1. INTRODUCTION

1.1. Overview

Customer churn is a major problem and one of the most important concerns for large companies. Due to the direct effect on the revenues of the companies, especially in the telecom field, companies are seeking to develop means to predict potential customer to churn. Therefore, finding factors that increase customer churn is important to take necessary actions to reduce this churn. Churn prediction helps in identifying those customers who are likely to leave a company. The main contribution of our work is to develop a churn prediction model which assists telecom operators to predict customers who are most likely subject to churn. The model developed in this work uses machine learning techniques on IBM platform and builds a new way of features' engineering and selection.

1.2. Purpose

Telecommunication industry always suffers from a very high churn rates when one industry offers a better plan than the previous there is a high possibility of the customer churning from the present due to a better plan in such a scenario it is very difficult to avoid losses but through prediction we can keep it to a minimal level. We are building a Machine Learning model to predict the customer churn using IBM Watson AutoAI Machine Learning Service. The model is deployed on IBM cloud to get scoring end point which can be used as API in mobile app or web app building. We are developing a web application which is built using node red service. We make use of the scoring end point to give user input values to the deployed model.
The model prediction is then showcased on User Interface.

2. LITERATURE SURVEY

2.1. Existing Problem

Churn rate is the percentage of subscribers to a service that discontinue their subscription to that service in a given time period. In order for a company to expand its clientele, its growth rate (i.e. its number of new **customers**) must exceed its **churn** rate.Customer churn is a major problem and one of the most important concerns for large companies. Due to the direct effect on the revenues of the companies, especially in the telecom field, companies are seeking to develop means to predict potential customer to churn.

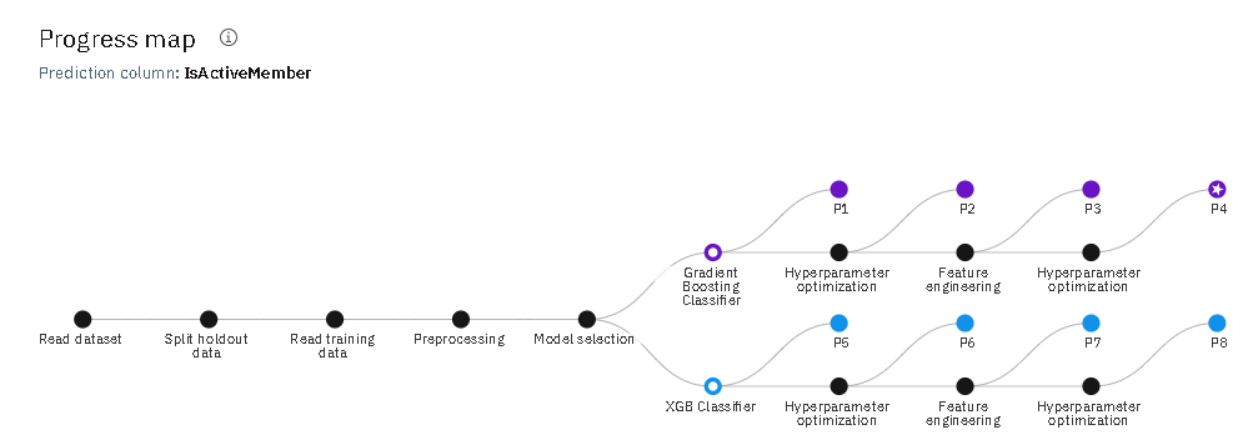
2.2. Proposed Solution

The telecommunications sector has become one of the main industries in developed countries. The technical progress and the increasing number of operators raised the level of competition.Customers' churn is a considerable concern in service sectors with high competitive services. On the other hand, predicting the customers who are likely to leave the company will represent potentially large additional revenue source if it is done in the early phase .We focused on evaluating and analyzing the performance of a set of tree-based machine learning methods and algorithms for predicting churn in telecommunications companies. We have experimented a number of algorithms such as Decision Tree, Random Forest, Gradient Boost Machine Tree and XGBoost tree to build the predictive model of customer Churn after developing our data preparation, feature engineering, and feature selection methods.We are building a Machine Learning model to predict the customer churn using IBM Watson AutoAI Machine Learning Service. The model is deployed on IBM cloud to get scoring end point which can be used as API in mobile app or web app building. We are developing a web application which is built using node red service. We make use of the scoring end point to give user input values to the deployed model.The model prediction is then

showcased on User Interface.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

For Auto AI solution:

1. Strategy: matching the problem with the solution.
2. Dataset preparation and pre-processing. Data collection.
3. Adding Dataset to the Watson Machine Learning.
4. Doing Auto AI analysis to find out the best model.
5. Model deployment.
6. Making Node Red flow.
7. Deploying the machine learning model through that Flow Application.

For own ipynb Notebook solution:

8. Strategy: matching the problem with the solution.
9. Dataset preparation and pre-processing. Data collection. Data visualization. Labelling. Data selection. Data pre-processing. Data transformation.
10. Dataset splitting into train data and test data.

11. Modelling. Model training. Model evaluation and testing. Improving predictions with ensemble methods.
12. Model deployment.
13. Making Node Red flow.
14. Deploying the machine learning model through that Flow Application.

4. EXPERIMENTAL INVESTIGATIONS

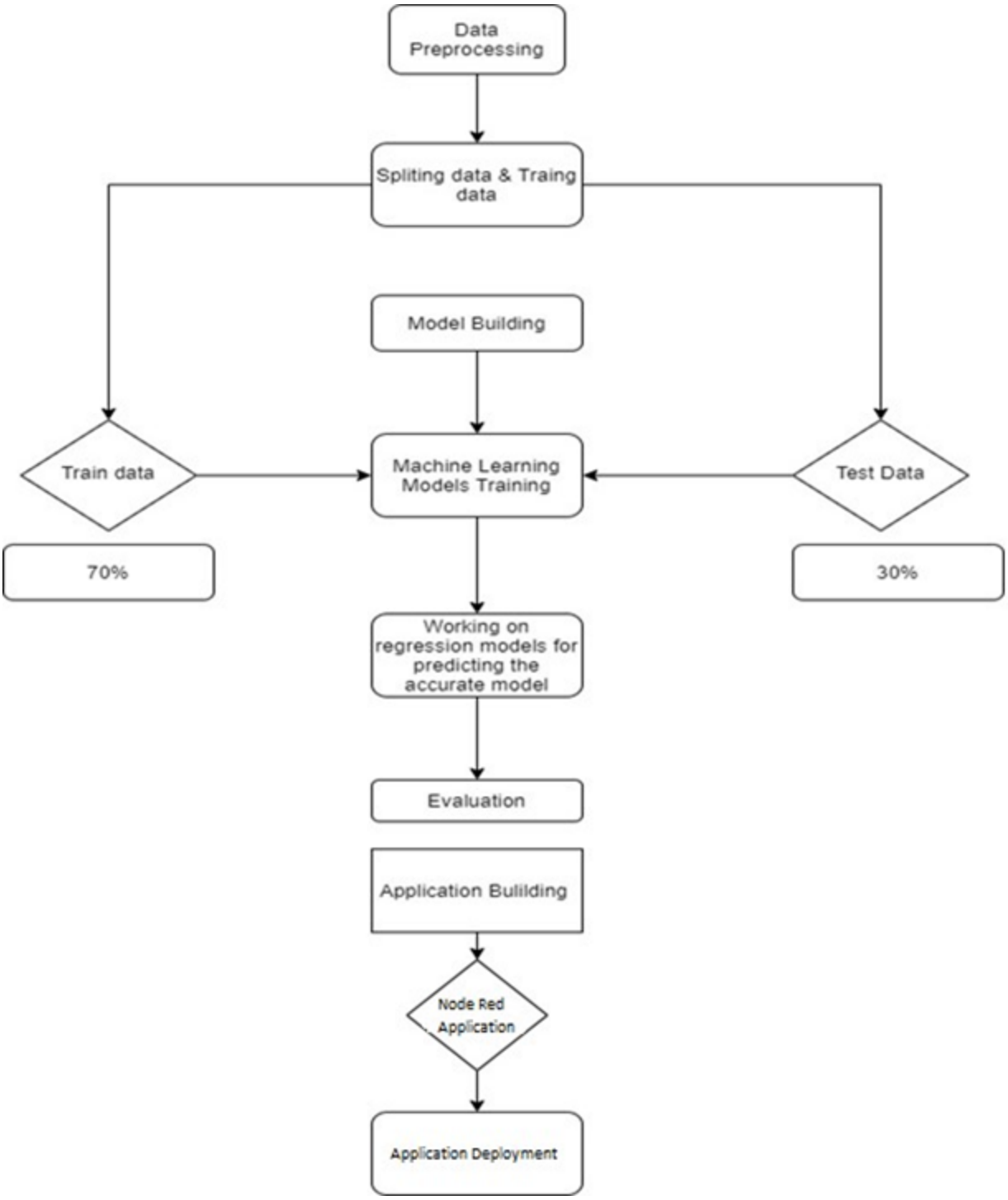
The isActive Member data for the present work was obtained from the experiments. This data set contains details of a bank's customers and the target variable is a binary variable reflecting the fact whether the customer left the bank (closed his account) or he continues to be a customer.

Various Parameters:

- RowNumber
- CustomerId
- Surname
- CreditScore
- Geography
- Gender
- Age
- Tenure
- Balance
- NumberOfProducts
- HasCreditCard
- Estimated Salary
- Exited

5. FLOWCHART

For own ipynb notebook model:


Data Preprocessing
Spliting data & Traing data
Model Building
Train data
Machine Learning Models Training
Test Data
70%
30%
Working on regression models for predicting the accurate model
Evaluation
Application Bulilding
Node Red Application
Application Deployment

6. RESULT

Based on the 13 inputs entered by the user, the model predicts the is Active Member and displays the predicted strength. And gives the output according to the entries in the Node red application.

7. ADVANTAGES & DISADVANTAGES

7.1. Advantages

1. Unlike traditional methods there is no wastage of test samples.
2. Higher accuracy can reduces errors in churn rates prediction.
3. Reduce the cost of finding out churn rate of customers.
4. Easy user interface with straight forward prediction.

7.2. Disa***dvantages***

1. The model is limited to predict the isActive Member for only companies which have exactly 13 inputs.
2. The Prediction basically holds on the input data which must be known at the time of Prediction.

8. APPLICATIONS

1. It can be used to predict the telecomm customer churn using several parameters.
2. Implementable on the website.
3. Can also be made into a phone app.

9. CONCLUSION

The importance of this type of research in the telecom market is to help companies make more profit. It has become known that predicting churn is one of the most important sources of income to telecom companies.Hence, this research aimed to build

a system that predicts the churn of customers in SyriaTel telecom company. These prediction models need to achieve high AUC values. To test and train the model, the sample data is divided into 70% for training and 30% for testing. We chose to perform cross-validation with 10-folds for validation and hyperparameter optimization. We have applied feature engineering, effective feature transformation and selection approach to make the features ready for machine learning algorithms. In addition, we encountered another problem: the data was not balanced. Only about 5% of the entries represent customers' churn. This problem was solved by undersampling or using trees algorithms not affected by this problem. Four tree based algorithms were chosen because of their diversity and applicability in this type of prediction. These algorithms are Decision Tree, Random Forest, GBM tree algorithm, and XGBOOST algorithm. The method of preparation and selection of features and entering the mobile social network features had the biggest impact on the success of this model, since the value of AUC in SyriaTel reached 93.301%. XGBOOST tree model achieved the best results in all measurements. The AUC value was 93.301%. The GBM algorithm comes in the second place and the random forest and Decision Tree came third and fourth regarding AUC values. We have evaluated the models by fitting a new dataset related to different periods and without any proactive action from marketing, XGBOOST also gave the best result with 89% AUC. The decrease in result could be due to the non-stationary data model phenomenon, so the model needs training each period of time.

The use of the Social Network Analysis features enhance the results of predicting the churn in telecom.

10. FUTURE SCOPE

Due to the direct effect on the revenues of the companies, especially in the telecom field, companies are seeking to develop means to predict potential customer to churn. Therefore, finding factors that increase customer churn is important to take necessary actions to reduce this churn. Churn prediction helps in identifyingthose customers who are likely to leave a company. The main contribution of our work is to develop a churn prediction model which assists telecom operators to predict customers who are most likely subject to churn. The model developed in this work uses machine learning techniques on IBM platform and builds a new way of features' engineering and selection.

11. BIBILOGRAPHY

11. BiBloGraphy

- https://smartbridge.teachable.com/courses/843009/lectures/17692123
- https://smartinternz.com/Student/workspace/3429
- https://www.kaggle.com/shrutimechlearn/churn-modelling

12 . APPENDIX

A.Source Code

Setup :

- watson-machine-learning-client uninstallation of the old client
- watson-machine-learning-client-V4 installation
- autoai-libs installation/upgrade
- lightgbm or xgboost installation/downgrade if they are needed

In [ ]:

```
!pip uninstall watson-machine-learning-client -y
```

In [ ]:

```
!pip install -U watson-machine-learning-client-V4
```

In [ ]:

```
!pip install -U autoai-libs
```

AutoAI experiment metadata

This cell defines COS credentials required to retrieve AutoAI pipeline.

In [ ]:

```
# @hidden_cell
from watson_machine_learning_client.helpers import DataConnection, S3Connection, S3Location
training_data_reference = [DataConnection(
  connection=S3Connection(
    api_key='7oYBJNYquzv5XS4sdQrceVIjskdkn0jSE5v5B3mPlFM5',
    auth_endpoint='https://iam.bluemix.net/oidc/token/',
    endpoint_url='https://s3-api.us-geo.objectstorage.softlayer.net'
  ),
    location=S3Location(
    bucket='predictionusingwatsonautoai-donotdelete-pr-jednjqnguwzeiu',
    path='Churn_Modelling.csv'
  ))
]
training_result_reference = DataConnection(
  connection=S3Connection(
    api_key='7oYBJNYquzv5XS4sdQrceVIjskdkn0jSE5v5B3mPlFM5',
    auth_endpoint='https://iam.bluemix.net/oidc/token/',
    endpoint_url='https://s3-api.us-geo.objectstorage.softlayer.net'
  ),
  location=S3Location(
    bucket='predictionusingwatsonautoai-donotdelete-pr-jednjqnguwzeiu',

path='auto_ml/efb44e5b-4332-417a-8e45-dd768e33d076/wml_data/99dd527f-14b8-410a-b9b4-842b1c570c7d/data/automl',

model_location='auto_ml/efb44e5b-4332-417a-8e45-dd768e33d076/wml_data/99dd527f-14b8-410a-b9b4-842b1c570c7d/data/automl/hpo_c_output/Pipeline1/model.pickle',

training_status='auto_ml/efb44e5b-4332-417a-8e45-dd768e33d076/wml_data/99dd527f-14b8-410a-b9b4-842b1c570c7d/training-status.json'
  ))
```

Following cell contains input parameters provided to run the AutoAI experiment in Watson Studio

In [ ]:

```
experiment_metadata = dict(
  prediction_type='classification',
  prediction_column='IsActiveMember',
  test_size=0.1,
  scoring='accuracy',
  csv_separator=',',
  excel_sheet=0,
```

```
  max_number_of_estimators=2,
  training_data_reference = training_data_reference,
  training_result_reference = training_result_reference)
pipeline_name='Pipeline_4'
```

Pipeline inspection

In this section you will get the trained pipeline model from the AutoAI experiment and inspect it.
You will see pipeline as a pythone code, graphically visualized and at the end, you will perform a local test.

Get historical optimizer instance

The next cell contains code for retrieving fitted optimizer.

In [ ]:

```
from watson_machine_learning_client.experiment import AutoAI
optimizer = AutoAI().runs.get_optimizer(metadata=experiment_metadata)
```

Get pipeline model

The following cell loads selected AutoAI pipeline model. If you want to get pure scikit-learn pipeline specify `as_type='sklearn'` parameter. By default enriched scikit-learn pipeline is returned `as_type='lale'`.

In [ ]:

```
pipeline_model = optimizer.get_pipeline(pipeline_name=pipeline_name)
```

Preview pipeline model as python code

In the next cell, downloaded pipeline model could be previewed as a python code.
You will be able to see what exact steps are involved in model creation.

In [ ]:

```
pipeline_model.pretty_print(combinators=False, ipython_display=True)
```

Visualize pipeline model

Preview pipeline model stages as graph. Each node's name links to detailed description of the stage.

In [ ]:

```
pipeline_model.visualize()
```

Read training and holdout data

Retrieve training dataset from AutoAI experiment as pandas DataFrame.

In [ ]:

```
training_df, holdout_df = optimizer.get_data_connections()[0].read(with_holdout_split=True)
train_X = training_df.drop([experiment_metadata['prediction_column']], axis=1).values
train_y = training_df[experiment_metadata['prediction_column']].values
test_X = holdout_df.drop([experiment_metadata['prediction_column']], axis=1).values
y_true = holdout_df[experiment_metadata['prediction_column']].values
```

Test pipeline model locally

Note: you can chose the metric to evaluate the model by your own, this example contains only a basic scenario.

In [ ]:

```
from sklearn.metrics import accuracy_score
predictions = pipeline_model.predict(test_X)
score = accuracy_score(y_true=y_true, y_pred=predictions)
print('accuracy_score: ', score)
```

Pipeline refinery and testing (optional)

In this section you will learn how to refine and retrain the best pipeline returned by AutoAI. It can be performed by:

- modifying pipeline definition source code
- using lale library for semi-automated data science

Note: In order to run this section change following cells to 'code' cell.

Pipeline definition source code

Following cell lets you experiment with pipeline definition in python, e.g. change steps parameters. It will inject pipeline definition to the next cell.

```
pipeline_model.pretty_print(combinators=False, ipython_display='input')
```

Lale library

Note: This is only an exemplary usage of lale package. You can import more different estimators to refine downloaded pipeline model.

Import estimators

```
from sklearn.linear_model import LogisticRegression as E1
from sklearn.tree import DecisionTreeClassifier as E2
from sklearn.neighbors import KNeighborsClassifier as E3
from lale.lib.lale import Hyperopt
from lale.operators import TrainedPipeline
from lale import wrap_imported_operators
from lale.helpers import import_from_sklearn_pipeline
wrap_imported_operators()
```

Pipeline decomposition and new definition

In this step the last stage from pipeline is removed.

```
prefix = pipeline_model.remove_last().freeze_trainable()
prefix.visualize()
new_pipeline = prefix >> (E1 | E2 | E3)
new_pipeline.visualize()
```

New optimizer hyperopt configuration and training

This section can introduce other results than the original one and it should be used by more advanced users.

New pipeline is re-trained by passing train data to it and calling `fit` method.

```
hyperopt = Hyperopt(estimator=new_pipeline, cv=3, max_evals=20)
fitted_hyperopt = hyperopt.fit(train_X, train_y)
hyperopt_pipeline = fitted_hyperopt.get_pipeline()
new_pipeline = hyperopt_pipeline.export_to_sklearn_pipeline()
predictions = new_pipeline.predict(train_X)
predictions = new_pipeline.predict(test_X)
refined_score = accuracy_score(y_true=y_true, y_pred=predictions)
print('accuracy_score: ', score)
print('refined_accuracy_score: ', refined_score)
```

Deploy and Score

In this section you will learn how to deploy and score pipeline model as webservice using WML instance.

Connect to WML client in order to create deployment

Action: Next you will need credentials for Watson Machine Learning and training run_id:

- go to Cloud catalog resources list
- click on Services and chose Machine Learning service. Once you are there
- click the **Service Credentials** link on the left side of the screen
- click to expand specific credentials name.
- copy and paste your WML credentials into the cell below

Take in mind that WML Service instance should be the same as used to generate this notebook.

In [ ]:

```
wml_credentials = {
  "apikey": "",
  "iam_apikey_description": "",
  "iam_apikey_name": "",
  "iam_role_crn": "r",
  "iam_serviceid_crn": "",
  "instance_id": "",
  "url": ""
}
```

Create deployment

Action: If you want to deploy refined pipeline please change the pipeline_model to new_pipeline.
If you prefer you can also change the deployment_name.

In [ ]:

```
from watson_machine_learning_client.deployment import WebService
service = WebService(wml_credentials)
service.create(
    model=pipeline_model,
    metadata=experiment_metadata,
```

```
    deployment_name=f'{pipeline_name}_webservice'
)
```

Deployment object could be printed to show basic information:

In [ ]:

```
print(service)
```

To be able to show all available information about deployment use .get_params() method:

In [ ]:

```
service.get_params()
```

Score webservice

You can make scoring request by calling score() on deployed pipeline.

In [ ]:

```
predictions = service.score(payload=holdout_df.drop([experiment_metadata['prediction_column']], axis=1).iloc[:10])
predictions
```

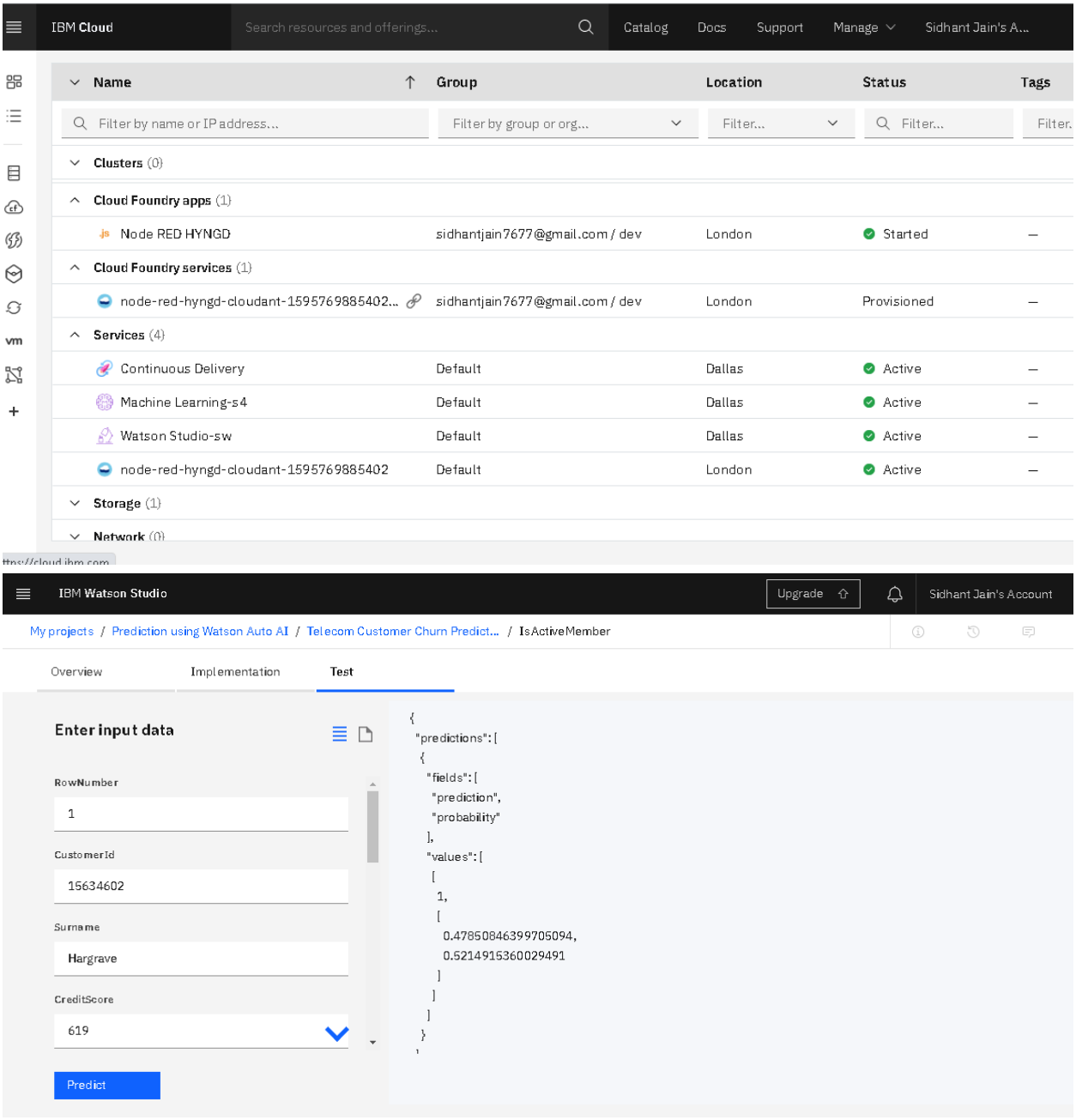
If you want to work with the webservice in external Python application you can retrieve the service object by:

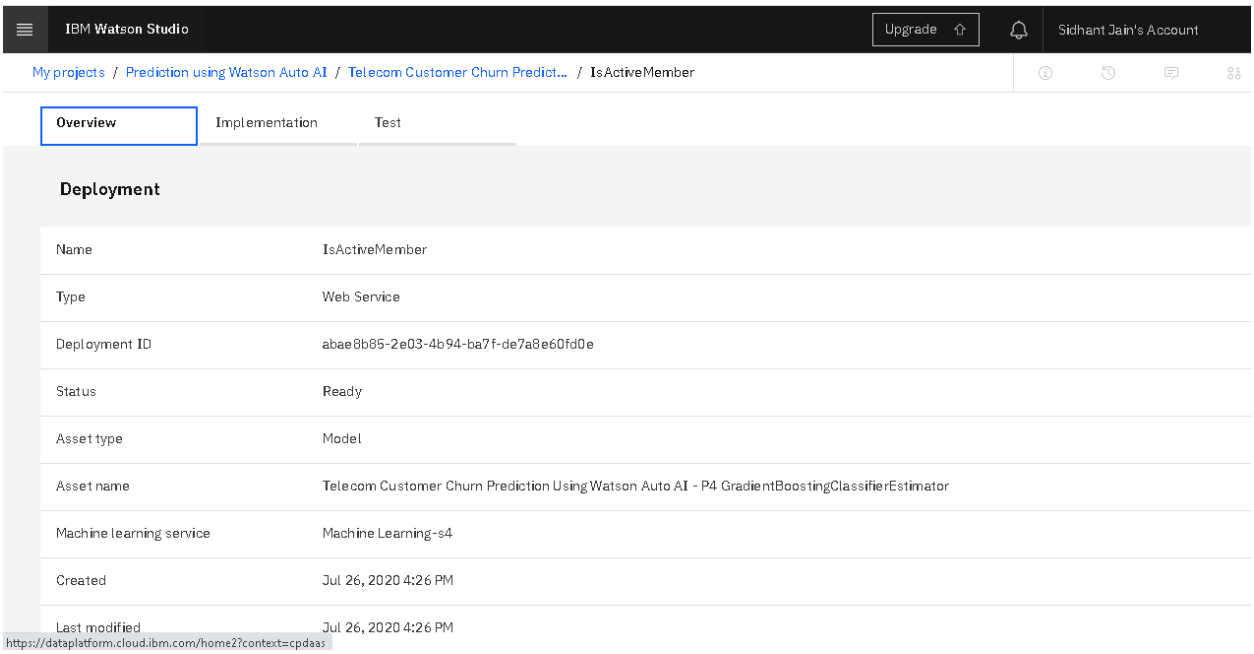
- initialize service by service = WebService(wml_credentials)
- get deployment_id by service.list() method
- get webservice object by service.get('deployment_id') method

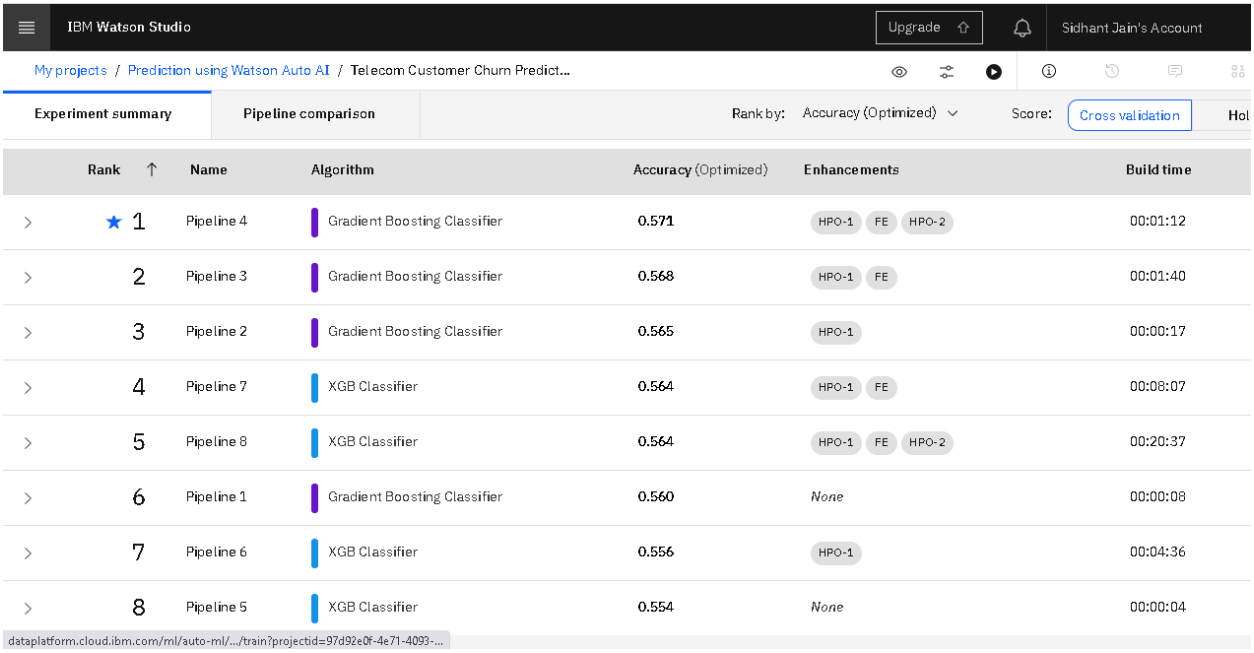
After that you can call service.score() method.

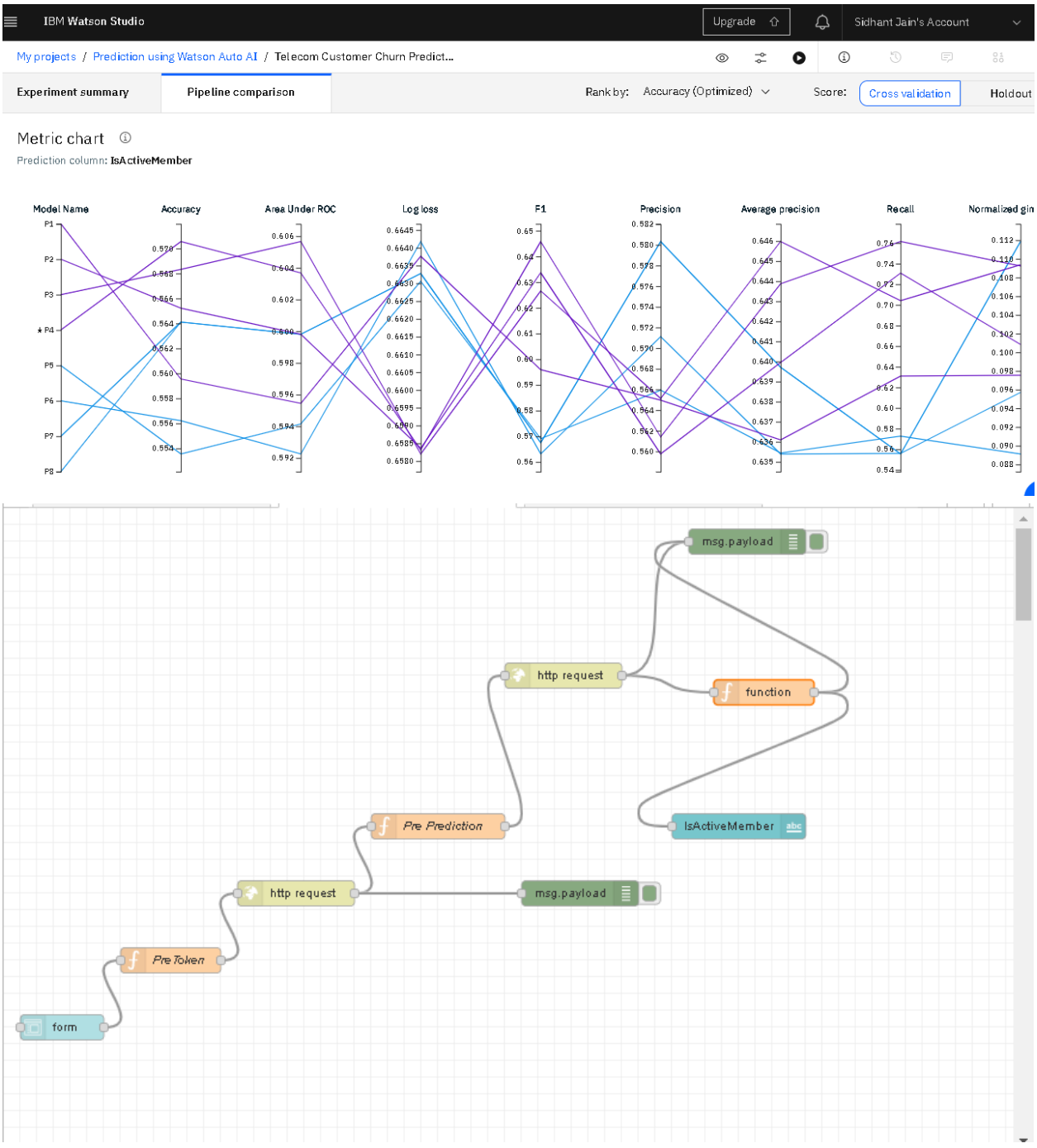
Delete deployment

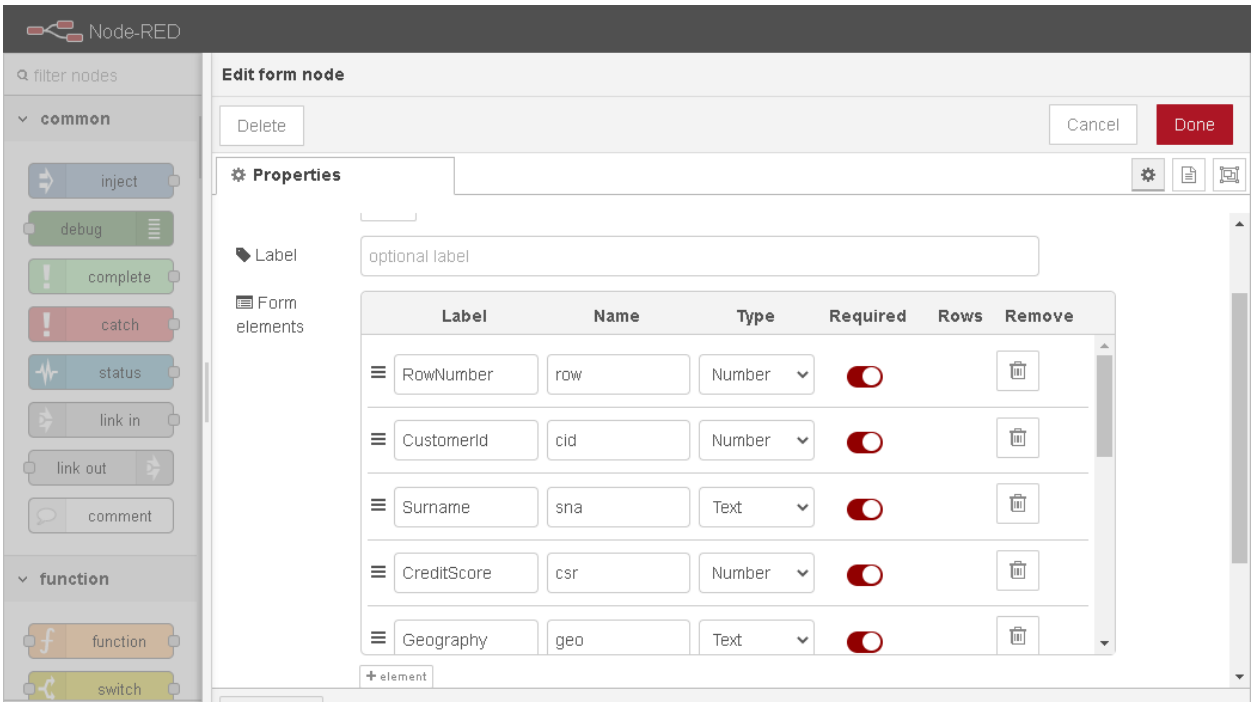
You can delete an existing deployment by calling service.delete().

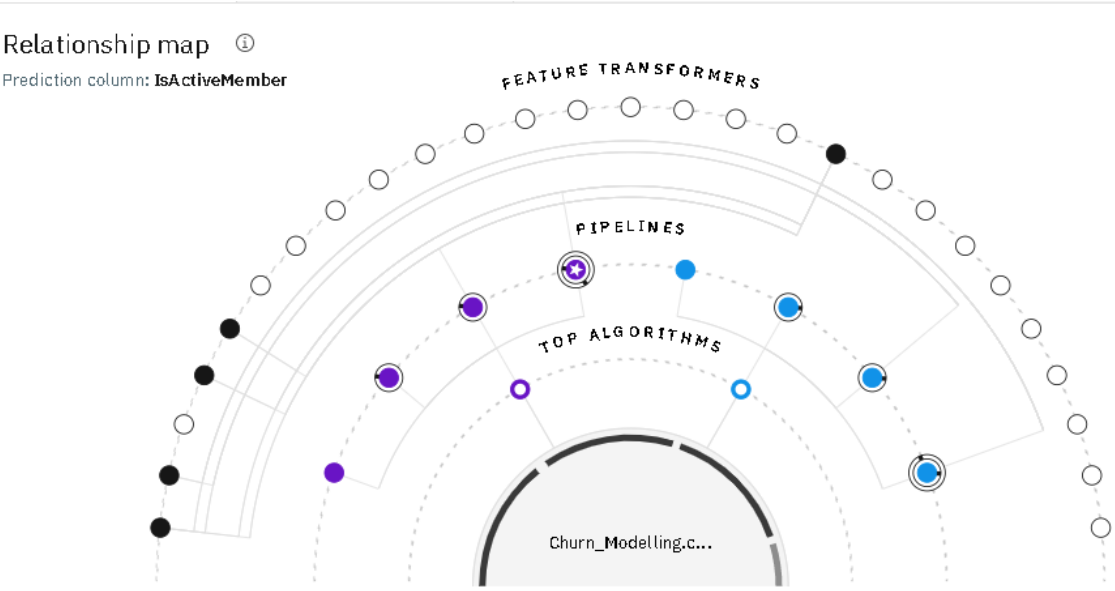

IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Sidhant Jain's A...
Name
Group
Location
Status
Tags
Filter by name or IP address...
Filter by group or org...
Filter...
Filter...
Clusters (0)
Cloud Foundry apps (1)
Node RED HYNGD
sidhantjain7677@gmail.com / dev
London
Started
Cloud Foundry services (1)
node-red-hyngd-cloudant-1595769885402...
sidhantjain7677@gmail.com / dev
London
Provisioned
Services (4)
Continuous Delivery
Default
Dallas
Active
Machine Learning-s4
Default
Dallas
Active
Watson Studio-sw
Default
Dallas
Active
node-red-hyngd-cloudant-1595769885402
Default
London
Active
Storage (1)
Network (0)
IBM Watson Studio
Upgrade
Sidhant Jain's Account
My projects / Prediction using Watson Auto AI / Telecom Customer Churn Predict... / IsActiveMember
Overview
Implementation
Test
Enter input data
RowNumber
1
CustomerId
15634602
Surname
Hargrave
CreditScore
619
Predict
{
"predictions": [
{
"fields": [
"prediction",
"probability"
],
"values": [
[
1,
[
0.47850846399705094,
0.5214915360029491
]
]
]
}
]


IBM Watson Studio
Upgrade
Sidhant Jain's Account
My projects / Prediction using Watson Auto AI / Telecom Customer Churn Predict... / IsActiveMember
Overview
Implementation
Test
Deployment
Name
IsActiveMember
Type
Web Service
Deployment ID
abae8b85-2e03-4b94-ba7f-de7a8e60fd0e
Status
Ready
Asset type
Model
Asset name
Telecom Customer Churn Prediction Using Watson Auto AI - P4 GradientBoostingClassifierEstimator
Machine learning service
Machine Learning-s4
Created
Jul 26, 2020 4:26 PM
Last modified
Jul 26, 2020 4:26 PM
https://dataplatform.cloud.ibm.com/home2?context=cpdaas

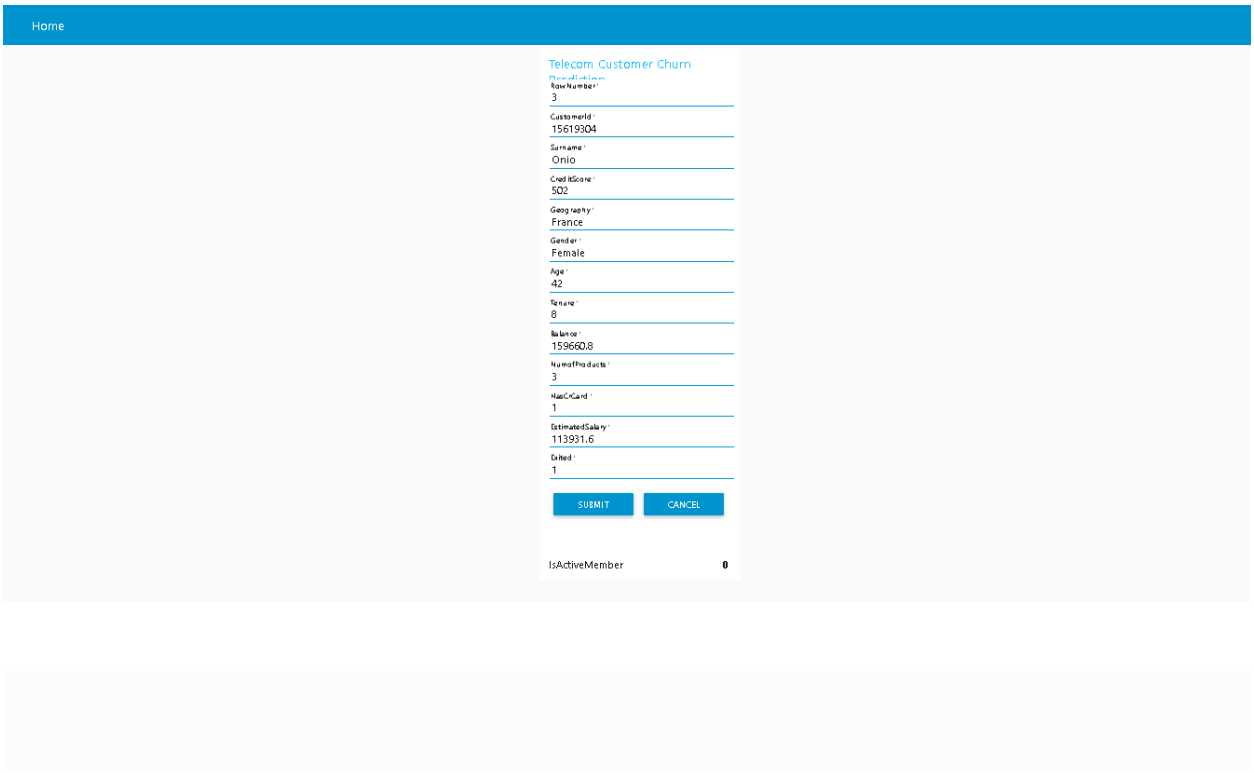

IBM Watson Studio
Upgrade
Sidhant Jain's Account
My projects / Prediction using Watson Auto AI / Telecom Customer Churn Predict...
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized)
Score: Cross validation
Rank
Name
Algorithm
Accuracy (Optimized)
Enhancements
Build time
1 Pipeline 4 Gradient Boosting Classifier 0.571 HPO-1 FE HPO-2 00:01:12
2 Pipeline 3 Gradient Boosting Classifier 0.568 HPO-1 FE 00:01:40
3 Pipeline 2 Gradient Boosting Classifier 0.565 HPO-1 00:00:17
4 Pipeline 7 XGB Classifier 0.564 HPO-1 FE 00:08:07
5 Pipeline 8 XGB Classifier 0.564 HPO-1 FE HPO-2 00:20:37
6 Pipeline 1 Gradient Boosting Classifier 0.560 None 00:00:08
7 Pipeline 6 XGB Classifier 0.556 HPO-1 00:04:36
8 Pipeline 5 XGB Classifier 0.554 None 00:00:04
dataplatform.cloud.ibm.com/ml/auto-ml/.../train?projectid=97d92e0f-4e71-4093-...


IBM Watson Studio
Upgrade
Sidhant Jain's Account
My projects / Prediction using Watson Auto AI / Telecom Customer Churn Predict...
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized)
Score: Cross validation
Holdout
Metric chart
Prediction column: IsActiveMember
Model Name
Accuracy
Area Under ROC
Log loss
F1
Precision
Average precision
Recall
Normalized gin
P1
P2
P3
* P4
P5
P6
P7
P8
msg.payload
http request
function
Pre Prediction
IsActiveMember
http request
msg.payload
Pre Token
form





B. UI Output Screenshot.


Home
Telecom Customer Churn
RowNumber *
3
CustomerId *
15619304
Surname *
Onio
CreditScore *
502
Geography *
France
Gender *
Female
Age *
42
Tenure *
8
Balance *
159660.8
NumOfProducts *
3
HasCrCard *
1
EstimatedSalary *
113931.6
Exited *
1
SUBMIT
CANCEL
IsActiveMember
0