

# Report for homework3

## Regression:

Deploy webservice for regression part linear regression, Random Forest and Neural Network.

### Linear Regression:

Using Notebook to create webservice:

Reading dataset and train linear regression model:

```
In [71]: model = linear_model.LinearRegression()
model.fit(X, y)
```

```
In [79]: workspace_id = ws.workspace_id
authorization_token = ws.authorization_token

from azureml import services
#services.service(workspace_id, authorization_token)
#services.service(workspace_id)
#services.types(orig_loantovalue=float,
#    occupancy_status_I=float,
#    credit_score=float,
#    orig_upb=float, channel_T=float,
#    loan_purpose_N=float, channel_R=float,
#    no_borrower=float,
#    orig_debttoincome=float,
#    occupancy_status_S=float,
#    first_time_homebuyer_flag_Y=float,
#    prepayment_penalty_mortgage_flag_Y=float,
#    loan_purpose_P_no_unit=float,
#    occupancy_status_O=float,
#    loan_purpose_C=float)
#services_returns(float)
#def service(orig_loantovalue,occupancy_status_I,
#    credit_score,orig_upb,channel_T,loan_purpose_N,channel_R,no_borrower,
#    orig_debttoincome,occupancy_status_S,first_time_homebuyer_flag_Y,
#    prepayment_penalty_mortgage_flag_Y,loan_purpose_P_no_unit,occupancy_status_O,loan_purpose_C):
#    # predict the label
#    feature_vector = [
#        orig_loantovalue,
#        occupancy_status_I,
#        credit_score,
#        orig_upb,
#        channel_T,
#        loan_purpose_N,channel_R,
#        no_borrower,
#        orig_debttoincome,
#        occupancy_status_S,
#        first_time_homebuyer_flag_Y,
#        prepayment_penalty_mortgage_flag_Y,
#        loan_purpose_P_no_unit,
#        occupancy_status_O,
#        loan_purpose_C]
#
#    return model.predict(feature_vector)
#/home/nbuser/anaconda2_20/lib/python2.7/json/encoder.py:207: DeprecationWarning: Interpreting naive datetime as local
2017-08-01 20:36:37.671462. Please add timezone info to timestamps.
chunks = self.iterencode(o, _one_shot=True)
```

Visit this web service through url and api\_key:

```
In [78]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["channel_T", "no_unit", "no_borrower", "orig_loantovalue", "channel_R", "credit_sco
            "Values": [[ "56.0", "0.0", "699.0", "190000.0", "0.0", "1.0", "2.0", "42.0", "0.0", "0.0",
            "GlobalParameters": {}
        }
    }
body = str.encode(json.dumps(data))

url = 'https://usouthcentral.services.azureml.net/workspaces/0d32a72a10464c6d9d1d1929a701f9ae/services/81c9dc4e9c040b
api_key = 'FQhJxplV1S3lp1dp/rzR2BQc07Sm/MohXG8h+4U1S8keyjF86fXKVvt3ytgDmGanju1K859V2P+TDxWd1Xw=='
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)
    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

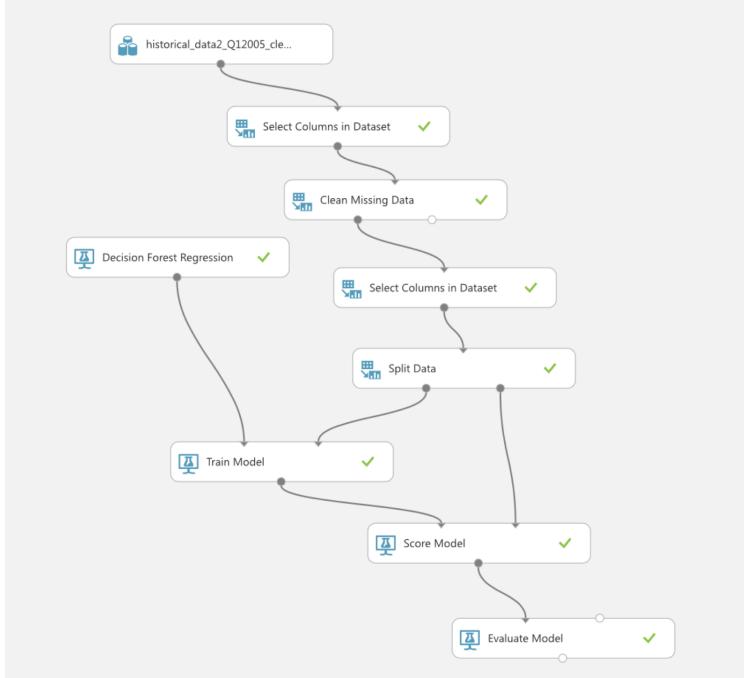
    # Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
    print(error.info())

    print(json.loads(error.read()))

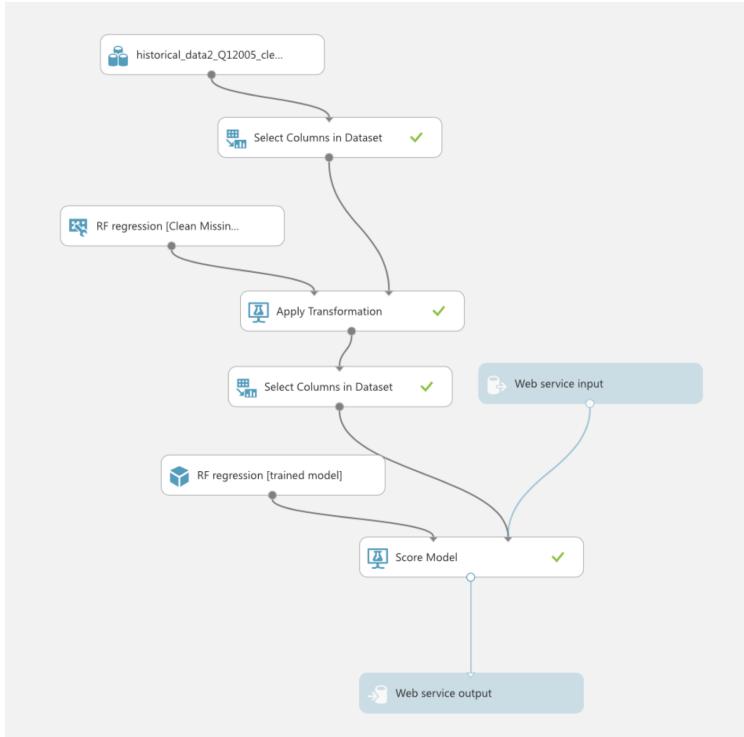
{ "Results": { "output1": { "type": "Table", "value": { "ColumnNames": [ " " ], "ColumnTypes": [ "Double" ], "Values": [ [ "847.647082910
578" ] ] }, "output2": { "type": "Table", "value": { "ColumnNames": [ "Standard Output", "Graphics" ], "ColumnTypes": [ "String" ], "Values": [ [ "data:text/plain,Execution OK\r\n" ] ] } } }}
```

## Random Forest:

### Data preprocessing and train and test model:



### Publish webservice:



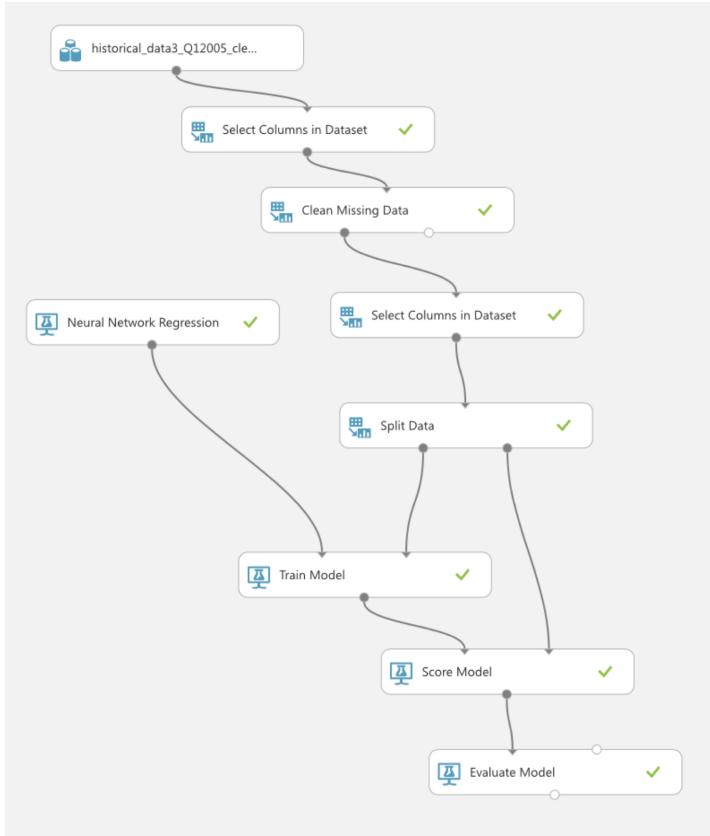
## Evaluate Random forest model:

RF regression ➤ Evaluate Model ➤ Evaluation results

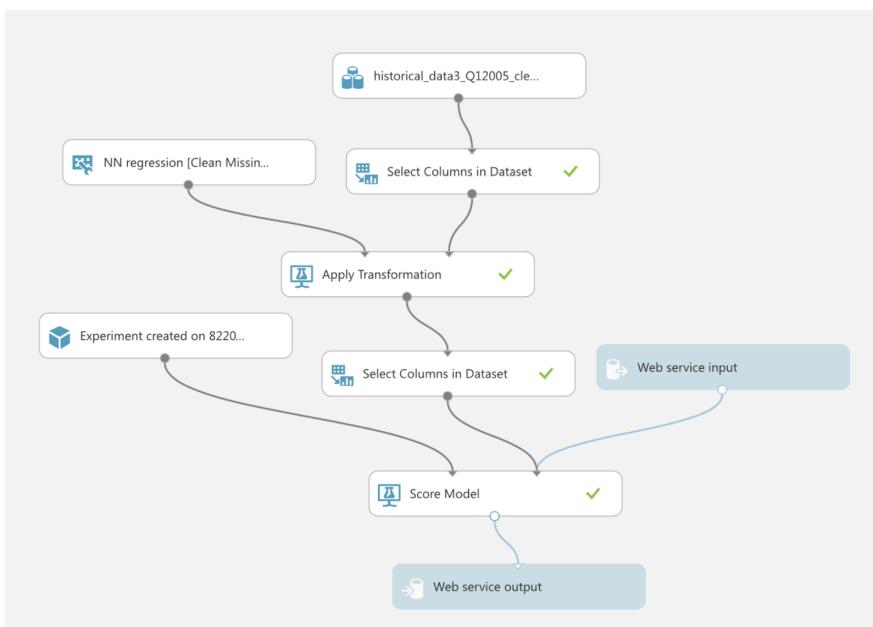
rows	columns
1	6
<hr/>	
Negative Log Likelihood	Mean Absolute Error
Root Mean Squared Error	Relative Absolute Error
Relative Squared Error	Coefficient of Determination
<hr/>	
view as	
 	
47172.311026	0.274812
0.35682	0.973621
0.945067	0.054933

## Neural Network regression:

Data preprocessing and train and test model:



Publish webservice:



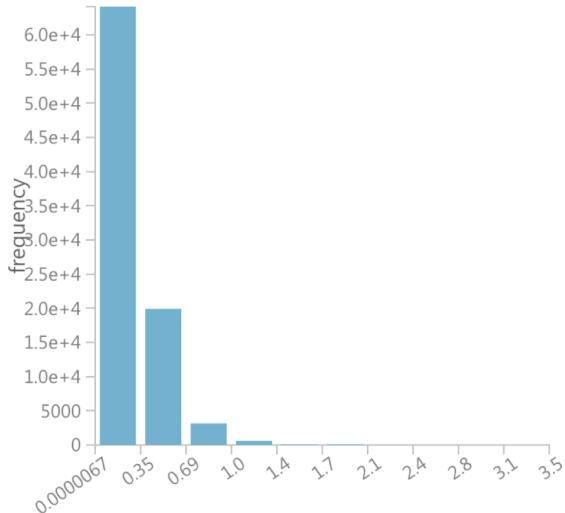
## Evaluate Random forest model:

NN regression ➤ Evaluate Model ➤ Evaluation results

Mean Absolute Error	0.2631
Root Mean Squared Error	0.340107
Relative Absolute Error	0.932129
Relative Squared Error	0.858611
Coefficient of Determination	0.141389

### ▲ Error Histogram

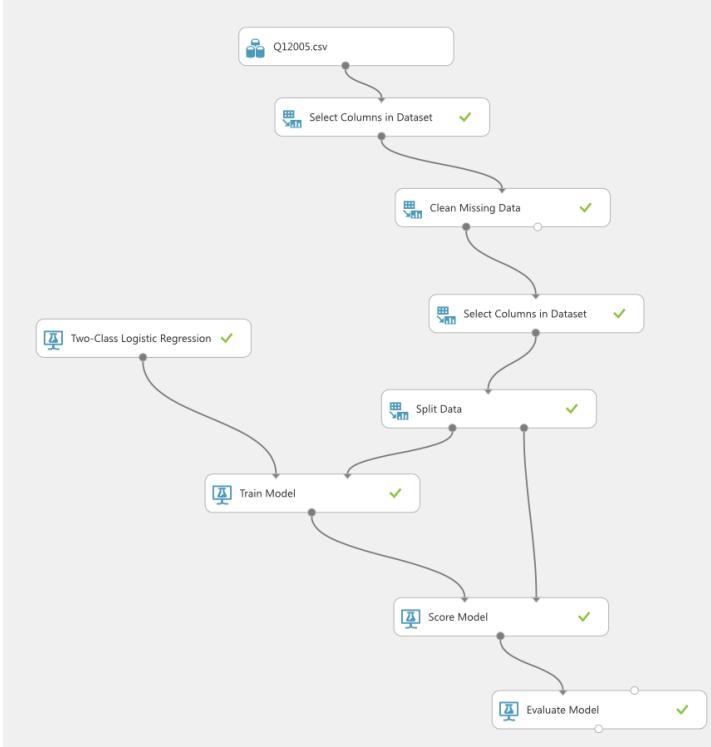
---



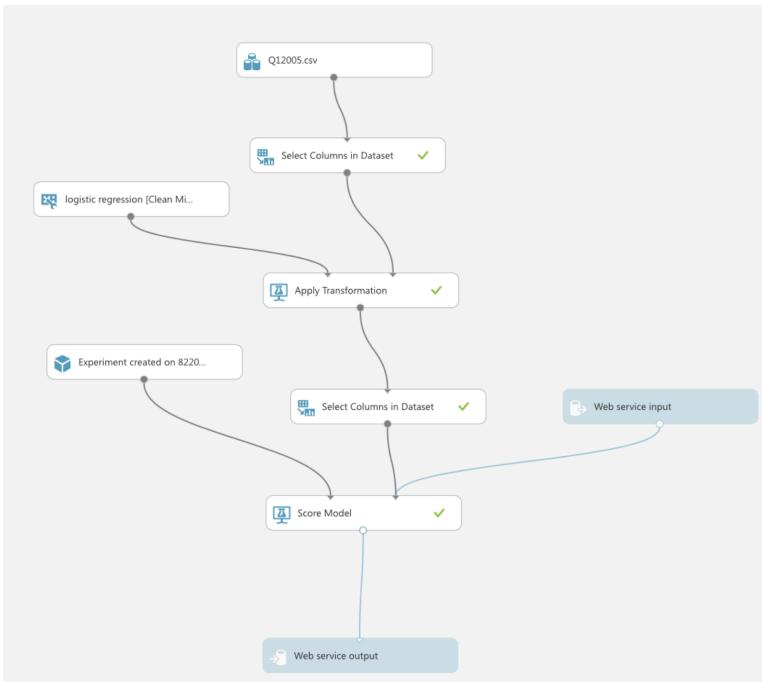
# Classification:

## Logistic regression:

Data preprocessing and train and test model:



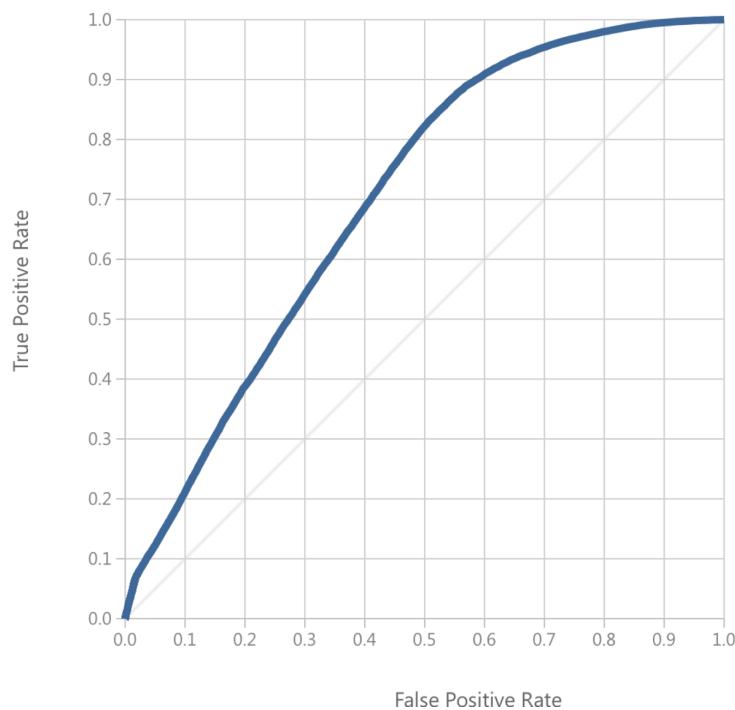
Publish webservice:



## Evaluate Random forest model:

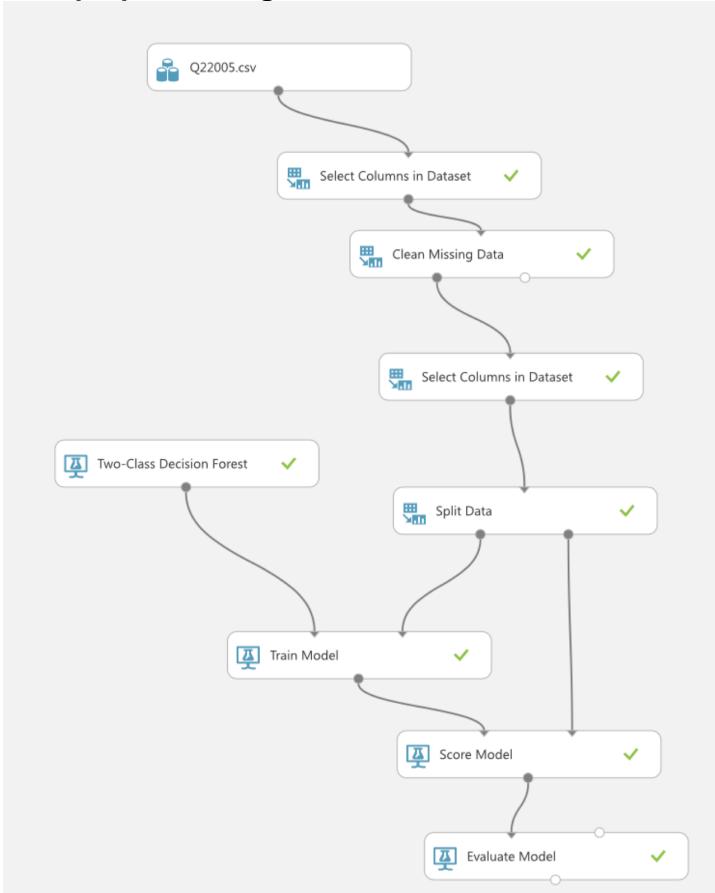
logistic regression ➤ Evaluate Model ➤ Evaluation results

ROC PRECISION/RECALL LIFT

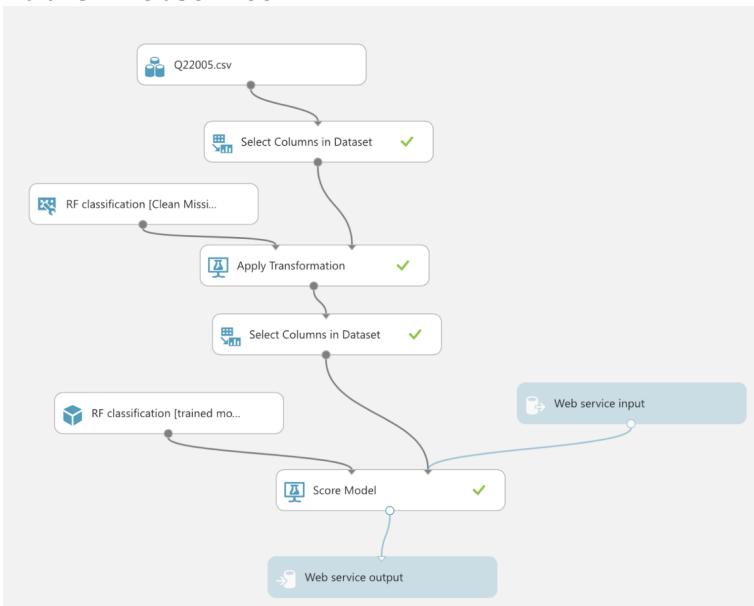


## Random Forest:

Data preprocessing and train and test model:



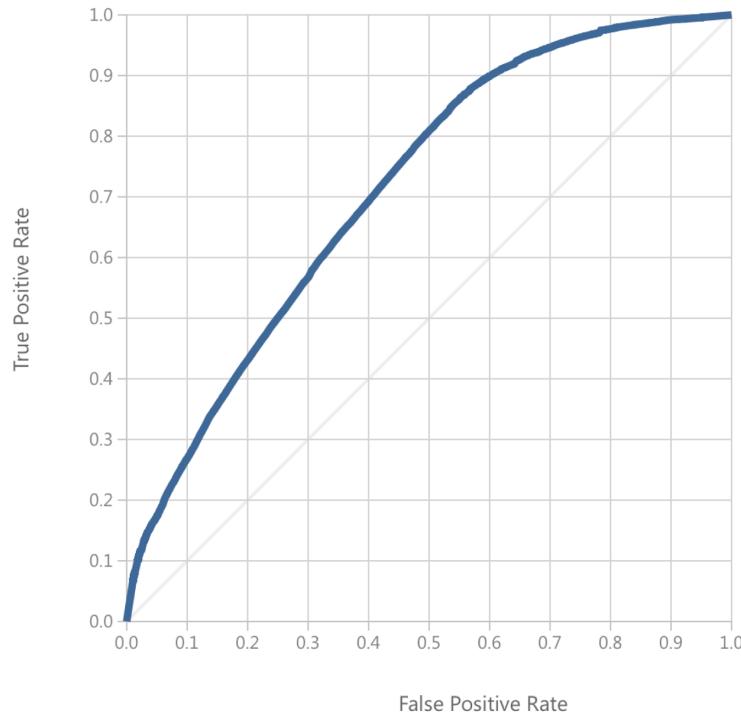
Publish webservice:



## Evaluate Random forest model:

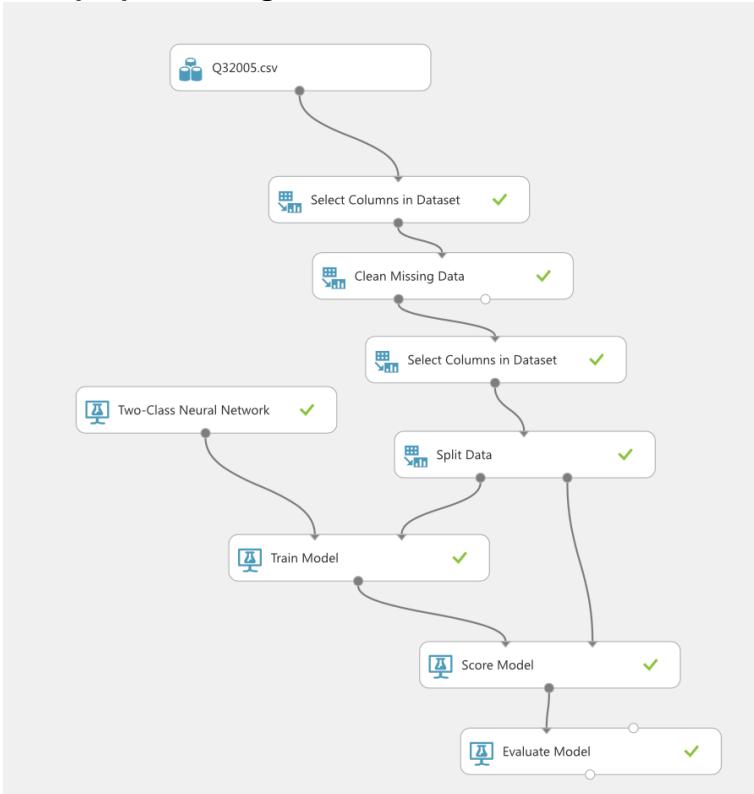
RF classification ➤ Evaluate Model ➤ Evaluation results

ROC PRECISION/RECALL LIFT

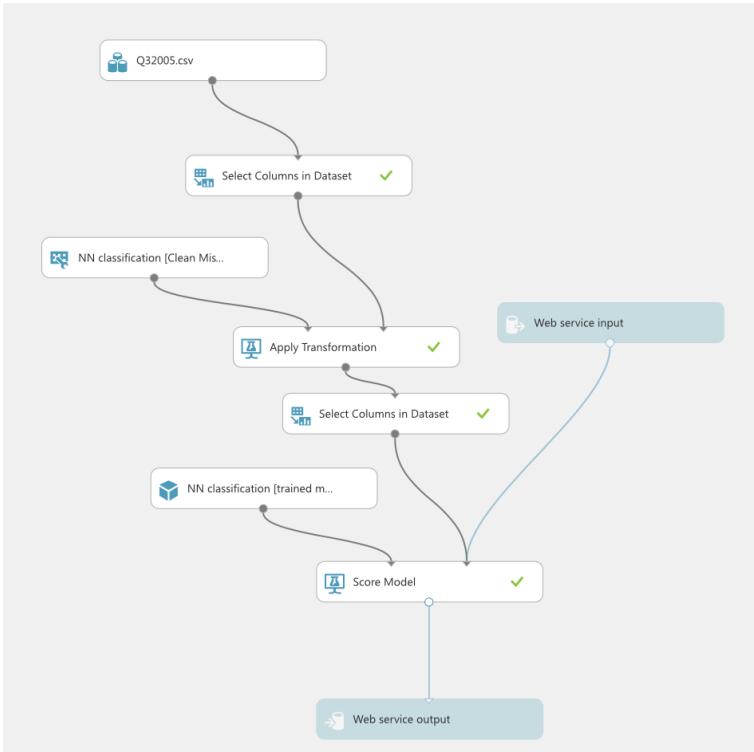


# Neural Network:

Data preprocessing and train and test model:



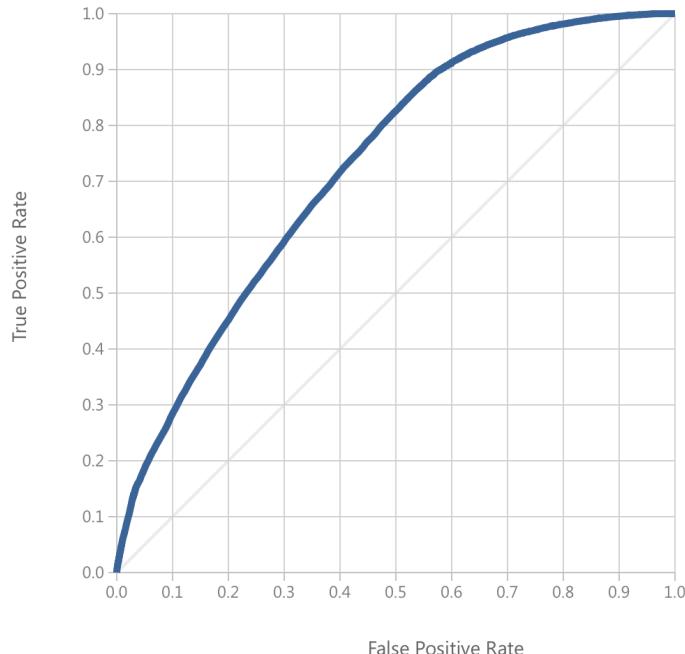
Publish webservice:



## Evaluate Random forest model:

NN classification ➤ Evaluate Model ➤ Evaluation results

ROC PRECISION/RECALL LIFT



## using notebook in my local system to test web service for all these 6 models

### Regression

#### Linear regression

```
In [2]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["channel_T", "no_unit", "no_borrower", "orig_loantoval",
                            "Values": [ [ "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0" ] ],
            "GlobalParameters": {}
        }
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/0d32a72e10464c6d9d1d1929a701
api_key = 'fQh+J1xpV1s3SpjdP/rZERCo7Tm/MobXG8hT+4UoISReyjFR6fxKEYvvt3ytqdEGanjulKH59V2P+T
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the timestamp, which are useful
    print(error.info())

    print(json.loads(error.read()))
```

#### Random Forest

```
In [4]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["credit_score", "first_time_homebuyer_flag", "no_unit",
                            "Values": [ [ "0", "value", "0", "value", "0", "0", "0", "value",
                                         "0", "value", "0", "value" ] ],
            "GlobalParameters": {}
        }
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/19e3e5bdfa6b42d381df6c8b26ce
api_key = 'h6Q9obInvZfdSb2G/39R/Ls8gQPScjdBVAZVT04zKKeilWzdXkFg6yN0F2Z+A8b4Hd0gP0B9pg0qn
headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the timestamp, which are useful
    print(error.info())

    print(json.loads(error.read()))
```

```
{"Results":{"output1":{"type":"table","value":{"ColumnNames":["credit_score","first_time
_homebuyer_flag","no_unit","occupancy_status","orig_loantoval","orig_loantoval
_overdue","orig_loantoval_purpose","orig_loantoval_purpose","orig_loantoval
_purpose","no_borrower","Scored Label Mean","Scored Label Standard Deviation"],"Colum
nTypes":["String","Int32","String","Int32","Double","String","String","String","String
","Int32","String","Int32","Int32","Int32","Double","String","String","String","String
","Int32","String","Double","Double"],"Values":[[{"0","value","0","value","0","0","0","va
lue","0","value","0","5.50661458333333","0.479252184646778"}, {"0","value","0","value
","0","0","0","value","0","value","0","5.50661458333333","0.479252184646778"]]}}}
```

## Neural Network

```
In [21]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["credit_score", "first_time_homebuyer_flag", "no_unit",
                            "Values": [ [ "0", "value", "0", "value", "0", "0", "0", "value",
                                         ],
                                         ],
                                         "GlobalParameters": {}
        }
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/19e3e5bd6a6b42d381df6c8b26ce
api_key = 'syJ0ewJLGX7Bw6zGET63ndjPyz+hNyICfd3GS5dKbaKikGyBy6m0mZzlhBa7xCm+uKgXqfN5LruFz

headers = {'Content-Type': 'application/json', 'Authorization':('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the timestamp, which are useful
    print(error.info())
    print(json.loads(error.read()))

{"Results":{"output1":{"type":"table","value":{"ColumnNames":["credit_score","first_time_homebuyer_flag","no_unit","occupancy_status","orig_debttoincome","orig_upb","orig_loantovalue","orig_interest_rate","channel","prepayment_penalty_mortgage_flag","loan_purpose","no_borrower","Scored_Labels"],"ColumnTypes":[{"Int32","String","Int32","String","String","Int32","Double","String","String","String","Int32","Double"]}, "Values": [[{"0","value","0","0","0","0","0","value","value","0","13.2027320861816"}, {"0","value","0","value","0","0","0","value","value","0","13.2027320861816"]]}}}}}
```

## Classification

### Logistic regression

```
In [1]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["curr_loan_delinquency_status", "loan_age", "remaining_months_to_legal_maturity", "curr_interest_rate", "Scored_Labels", "Scored_Probabilities"], "ColumnTypes": [{"Int32", "Int32", "Int32", "Double", "String", "String"}], "Values": [ [ 0, 0, 0, 0 ], [ 0, "0", "0", "0" ] ], "GlobalParameters": {}
        }
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/19e3e5bd6a6b42d381df6c8b26ce
api_key = 'ssSMAD1501G25WsrZRFN8tpo0/Oao/XKcawwIeQeIW/RXyEB1E8cPMtytbIHZMxpWXE7WekyxdhTKI+
headers = {'Content-Type': 'application/json', 'Authorization':('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the timestamp, which are useful
    print(error.info())
    print(json.loads(error.read()))

{"Results":{"output1":{"type":"table","value":{"ColumnNames":["curr_loan_delinquency_status","loan_age","remaining_months_to_legal_maturity","curr_interest_rate","Scored_Labels","Scored_Probabilities"],"ColumnTypes":[{"Int32","Int32","Int32","Double","String","String"}], "Values": [[{"0","0","0","0","0.00509015005081892"}, {"0,"0","0","0","0.00509015005081892"}]}}}}}
```

### Random Forest

```
In [25]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["curr_loan_delinquency_status", "loan_age", "remaining",
                            "Values": [ [ "0", "0", "0" ], [ "0", "0", "0" ] ],
                            " },
            "GlobalParameters": {}
        }
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/19e3e5bdfa6b42d381df6c8b26ce
api_key = 'AtkHfNlH9Chh0Rw7xfVaHVVpNb0QrJmt0r5xxPjp52qv/Brdnn26CYVbw6IYHUnWXM3NHJk5pRd462
headers = {'Content-Type': 'application/json', 'Authorization':('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the timestamp, which are useful
    print(error.info())

    print(json.loads(error.read()))

{"Results":{"output1":{"type":"table","value":{"ColumnNames":["curr_loan_delinquency_st
atus","loan_age","remaining_months_to_legal_maturity","curr_interest_rate","Scored_Label
s","Scored_Probabilities"],"ColumnTypes":["Int32","Int32","Int32","Double","Int32","Doub
le"],"Values":[[["0","0","0","0","0"],[ "0","0","0","0","0"]]}}}}
```

### Neural Network

```
In [26]: import urllib2
# If you are using Python 3+, import urllib instead of urllib2

import json

data = {
    "Inputs": {
        "input1": {
            "ColumnNames": ["curr_loan_delinquency_status", "loan_age", "remaining",
                            "Values": [ [ "0", "0", "0" ], [ "0", "0", "0" ] ],
                            " },
            "GlobalParameters": {}
        }
    }
}

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/19e3e5bdfa6b42d381df6c8b26ce
api_key = 'gTCNaqerhS4dK0dGEd20gDC3ewIWLcsMAUjWyqdyyVi5M5seedLIWJwYwGXu48Z0mBJGkQcnEUawgGM
headers = {'Content-Type': 'application/json', 'Authorization':('Bearer ' + api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib2.HTTPError, error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the timestamp, which are useful
    print(error.info())

    print(json.loads(error.read()))

{"Results":{"output1":{"type":"table","value":{"ColumnNames":["curr_loan_delinquency_st
atus","loan_age","remaining_months_to_legal_maturity","curr_interest_rate","Scored_Label
s","Scored_Probabilities"],"ColumnTypes":["Int32","Int32","Int32","Double","Int32","Doub
le"],"Values":[[["0","0","0","0","0"],[ "0","0","0","0","0"]]}}}}
```

# Web API

Web app for these 6 web services. We are using **flask REST api** for web app. We use **AWS EC2 instance** as host for running our web app. We can visit web service through visiting ip address.

← → ⌂ ⓘ 35.167.59.55/prediction

## Please give your prediction model

Predict Interest rate :

- Linear Regression
- Random Forest
- Neural Network

Predict delinquent status :

- Logistic Regression
- Random Forest
- Neural Network

**Submit**

We can select any one of these 6 models to predict for us. For example, if we select Neural Network for predicting our interest rate, and we give our input:

← → ⌂ ⓘ 35.167.59.55/model

## Neural Network to predict Interest Rate

orig\_loantvalue :

credit\_score :

orig\_upb :

no\_borrower :

orig\_debttoincome :

no\_unit :

orig\_combined\_loantvalue :

occupancy\_status :

channel :

loan\_purpose :

first\_time\_homebuyer\_flag :

prepayment\_penalty\_mortgage\_flag :

**Submit**

we click submit button and we can get the prediction result:

← → ⌂ ⓘ 35.167.59.55/nnregression

The prediction result is: 7.09272861480713

We select logistic regression to predict delinquency status:

← → ⌂ ⓘ 35.167.59.55/model

### Logistic Regression predict delinquent status

loan\_age :

remaining\_months\_to\_legal\_maturity :

curr\_interest\_rate :

loan\_sequence\_no :

monthly\_reporting\_period :

curr\_actual\_upb :

curr\_deferred\_upb :

Submit

← → ⌂ ⓘ 35.167.59.55/logisticregression

The prediction result is:

You have 0.00509015005081892 chance to be delinquent