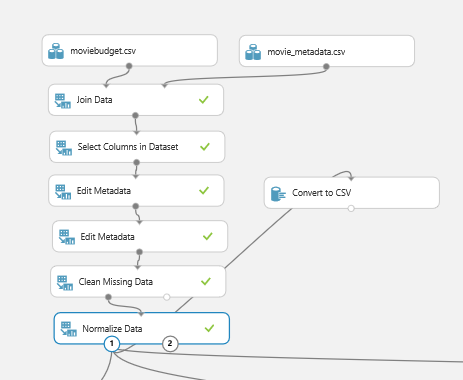
Normalize Data in Azure Machine Learning

1. If you have not already done so, open a browser and browse to https://studio.azureml.net. Then sign in using the Microsoft account associated with your Azure ML account.
2. Create a new blank experiment, and give it the title **MovieML Normalize (Python)**. Add your **moviebudget and movie\_metadata** saved datasets to the experiment canvas.
3. Find the **Join Data** module, and drag it to the experiment convas under two datasets. Select column names moviename for L, select movie\_title for R. select Inner join as Join type.
4. Find the **Select Columns in Dataset** module, and drag it to the experiment convas under Join Data. Exclude movie\_title, No., data columns.
5. Find the **Edit Metadata** module, and drag it to the experiment convas under Select Columns in Dataset. Include column Budget, domesticgross, globalgross and durantion. Then, create another **Edit Metadata** module, and drag it to the experiment convas under the first edit metadata.
6. Find the **Clean Missing Data** module, and drag it to the experiment convas under the second edit metadata module. Select column duration, language and country. Set the minimum missing value ratio to 0 and set maximum missing value ratio to 1. Select remove entire row for the cleaning mode.
7. Find the **Normalize Data** module, and drag it to the experiment convas under the clean missing data module. Select budget, domestigross, duration and imdb\_score.
8. Find the **Convert to CSV** module, and drag it to the experiment convas under the normalize Data. Right click this module and download to the local path.



Data Mining in Jupyter(python)

1. Open the Jupyter (Inotebook)
2. Type the following code in order to find out all the data we need

import pandas as pd

import numpy as np

from pandas import Series, DataFrame

#upload file

data = pd.read\_csv(r'C:\Users\teddy\Desktop\mlmovie1.csv')

genres = data['genres']

DataFrame(genres)

# split genres by using '|'

split = genres.apply(lambda x: Series(x.split('|')))

# show the first genres and combine with original data

new = data.join(split[0])

#delete the old genres column

dropcol = new.drop('genres', axis = 1)

#rename the new genres colum to majorgenres

rename = dropcol.rename(columns = {0:'majorgenres'})

#replace language column with numbers

lg = rename

lg['language'].replace(['English','French','Hindi','Spanish','Mandarin','Cantonese','German','Japanese','Russian','Korean',

'Arabic','Italian','Portuguese','Aboriginal','Polish','Dari','Dutch','Zulu','Hebrew','Telugu','Tamil',

'Swahili','Icelandic','Maya','Slovenian','Persian','Panjabi','Indonesian','Filipino','None','Swedish','Greek',

'Thai','Chinese','Norwegian','Vietnamese','Aramaic','Hungarian'],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,

19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,

34,35,36,37,38], inplace = True)

#replace country column with numbers

lg['country'].replace(['USA','UK','Canada','France','Germany','Australia','Spain','India','NewZealand', 'HongKong','Ireland','China','Japan','Italy','Mexico','SouthKorea','SouthAfrica', 'Denmark','Belgium','Russia','Switzerland','Netherlands','Thailand','Norway',

'Poland','CzechRepublic','Iceland','Israel','Hungary','Brazil','Greece','Romania','Bulgaria','Nigeria','Slovenia','Iran','NewLine','Aruba','Finland','Kenya','Bahamas','Georgia','Afghanistan','WestGermany','Egypt','Officialsite','SovietUnion','Chile','Colombia','Libya','Philippines','UnitedArabEmirates','Sweden','Panama','Peru','Cameroon'],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56], inplace = True)

#replace majorgenres column with numbers

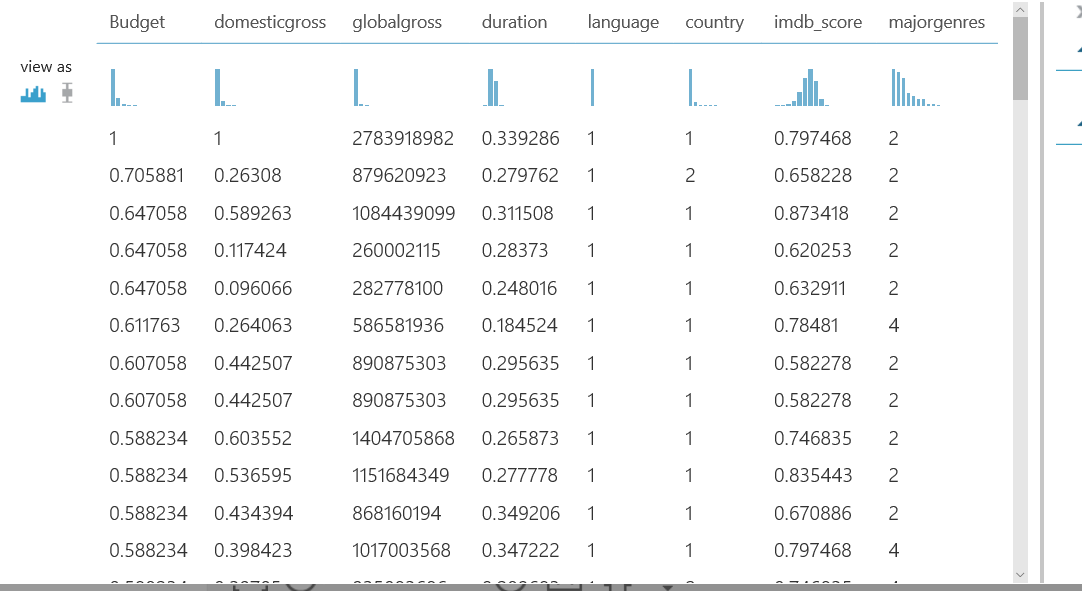
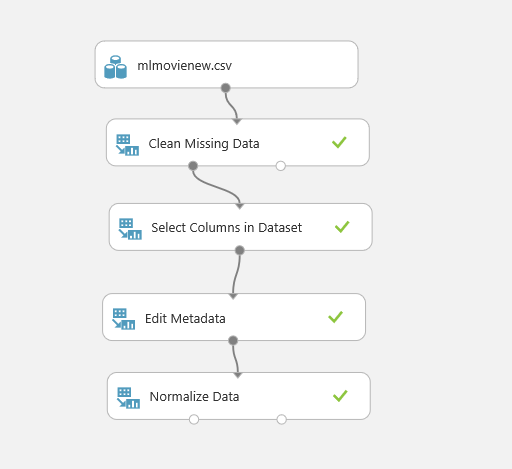
lg['majorgenres'].replace(['Comedy','Action','Drama','Adventure','Crime','Biography','Horror','Documentary','Fantasy','Animation','Mystery','Thriller','SciFi','Family','Western','Musical','Romance','FilmNoir','GameShow','Music'],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20], inplace = True)

#save file to local

rename.to\_csv(r'C:\Users\teddy\Desktop\mlmovienew.csv')

Clean Data in Azure Machine Learning

1. Create a new blank experiment, and give it the title **MovieML clean data (Python)**. Add your mlmovienew.csv into convas.
2. Find the **Clean Missing Data** module, and drag it to the experiment convas under mlmovienew.csv. Select language and country column, select remove entire row in cleaning mode.
3. Find the **Select Column in dataset** module, and drag it to the experiment convas under clean missing data module. Select column 0 and moviename columns.
4. Find the **Edit Metadata** module, and drag it to the experiment convas under select column in dataset module. Select language, country and majorgenres, set the categorical as make categorical.
5. Find the **Normalize Data** module, and drag it to the experiment convas under edit metadata module. Select budget, domesticgross, duration and imdb\_socre columns. Set transformation method as MinMax
6. Save and run the experiment. The experiment should looks like the following.



Evaluate Data in Azure Machine Learning

1. Find the **Split data** module, and drag it to the experiment convas under Normalize data module. Select the following:

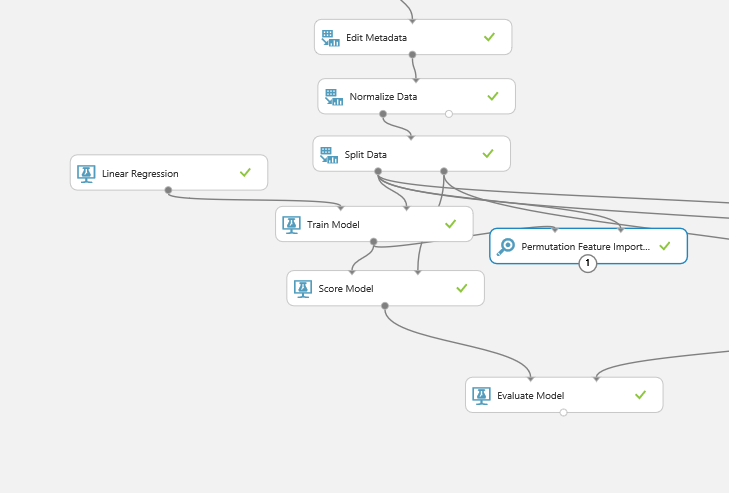
Splitting mode: Split Rows,

Fraction of rows: 0.7,

Random seed: 345

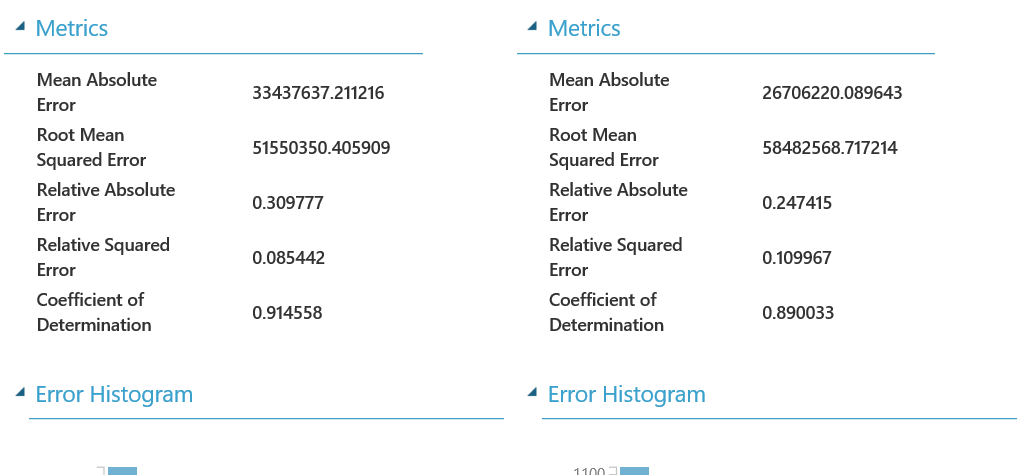
Stratified split: False

1. Find the **Linear Regression model**, and drag it to the experiment convas.
2. Find the **Train Model**, and drag it to the experiment convas, connect the linear regression module to the untrained model(left) of the train model, and connect the results dataset1 of split data to the dataset of Train Model.
3. Find the **Score Model**, and drag it to the experiment convas under the train model. Select globalgross column in the label column. Connect the top left dot to the train model and connect the dataset to the result dataset2 of split data.
4. Find the **Permutation Feature import module**, and drag it to the experiment convas and connect to the train model.
5. Find the **evaluation model** and drag it to the experiment convas under the score model.
6. Save and run the experiment, after done running. The experiment should look like following.



Compare two models

1. Copy the Train Model, Score Model, and paste on the right side of the experiment.
2. Find the Boosted Decision Tree Regression Model, and drag it to the experiment convas. Connect the BDT regression model with train model.
3. Connect the new score model to the right spot of the evaluation model.
4. Save and run the experiment, right click on the evaluate model, click visualize under the evaluation results, the results should be look like this.

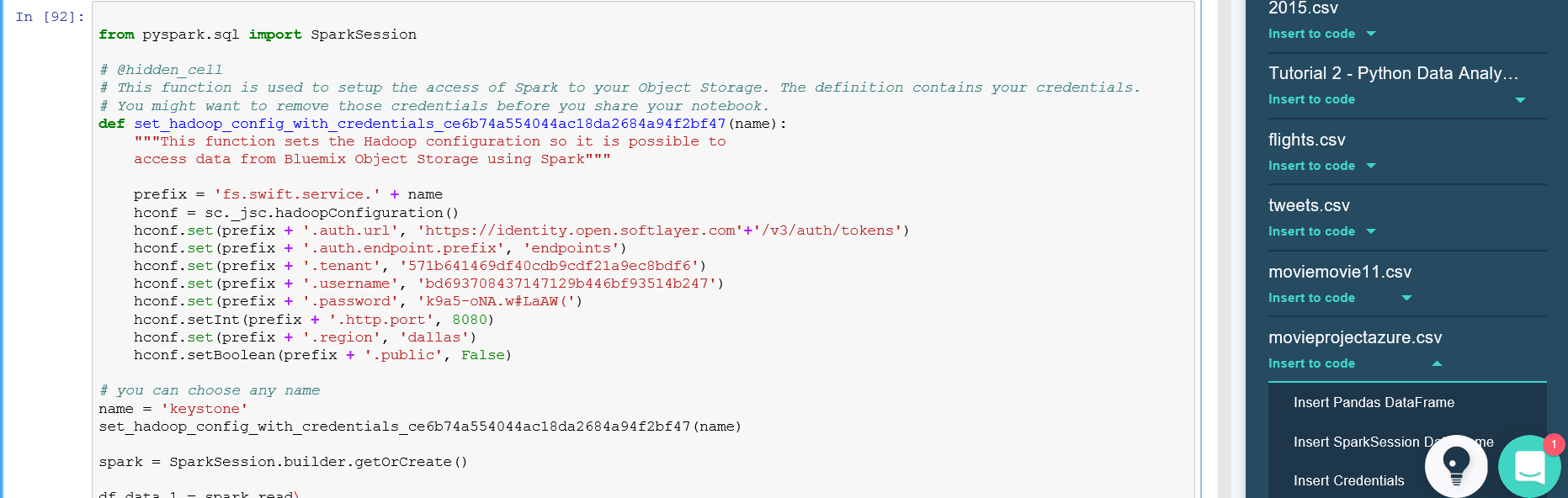


Sign in to the IBM Data Science Experiences

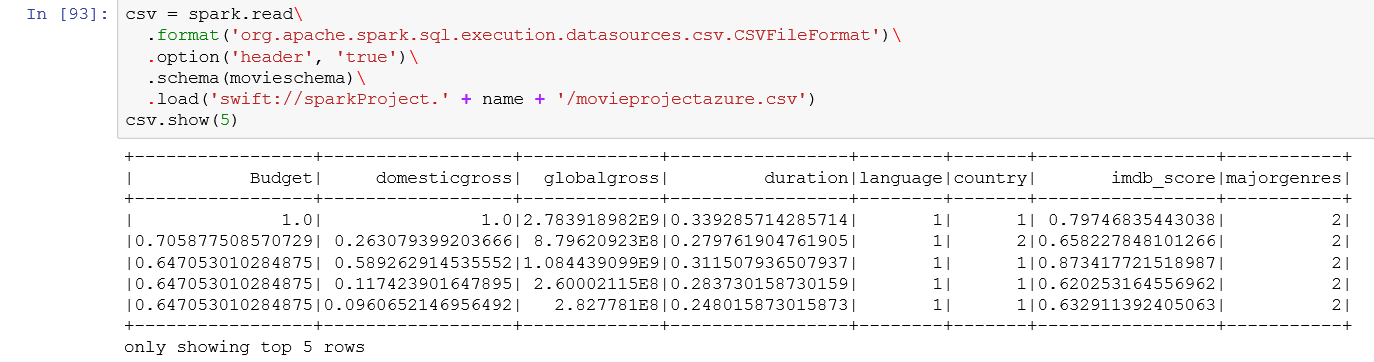
1. After you successfully log in, select the top right of the page and select “Create Project”. In the New Project page, you need to enter Name of the project, for example, sparkProject. Then, select “Create” button to move on the following page. Now you need to add Notebooks by clicking on “add notebooks”.
2. Then, create a blank notebook and enter the name. Then, click on Create Notebook button.
3. Now you have to add data source, that is, csv file to read and compute by this Spark code. Click on “Find and Add Data” icon in the top right. Once you see the frame in the tight side, click on “browse”.
4. Write the code as following to import the package we need and write the datatype we need.



1. You need to select **Insert Spark Session Setup** as the code given adopts Spark RDD. Then you will see that a code is automatically generated into the empty cell with your credentials to access and to read **movieprojectazure.csv** file.



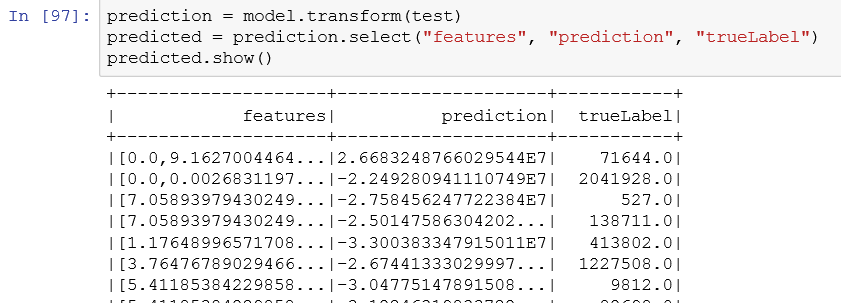
1. Write the code and visualize in the data frame.



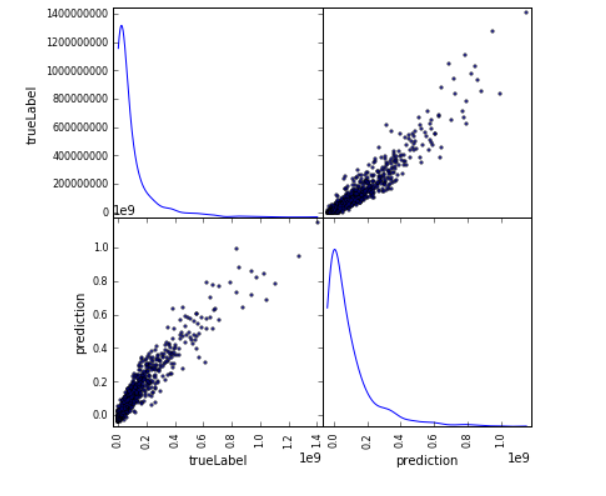
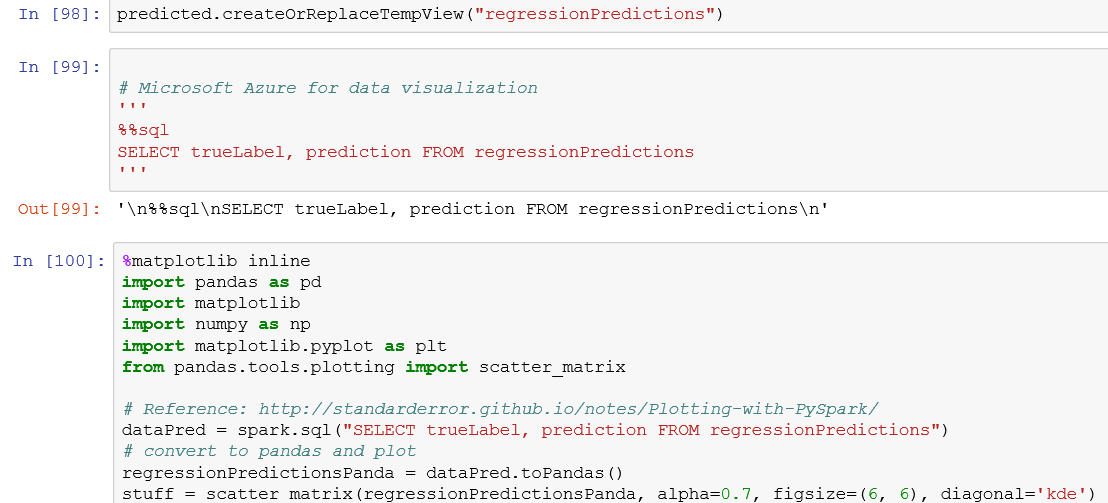
1. Select feature and label. Then, Split the data, Define the pipeline and train the data. The code should look like following.



1. Predict the value, the code and output should be look like the following.



1. Draw the graph, the code and graph should look like following.



1. Calculate the Root Mean Square Error. The result should look similar to this.

