# [https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS5560 Term Project Tutorial

#### Authors: [Kaustubh Padhya](https://www.linkedin.com/in/kaustubhpadhya), [Roshik Ganesan](https://www.linkedin.com/in/roshik-ganesan-925143a1), [Manali Joshi](https://www.linkedin.com/in/manali-joshi-2a2b9a100), [Mittal Vaghela](https://www.linkedin.com/in/mittal-vaghela-b2811177)

#### Instructor: Jongwook Woo

#### Date: 05/18/2017

**Lab Tutorial**

Kaustubh Padhya([kpadhya@calstatela.edu](mailto:kpadhya@calstatela.edu)), Roshik Ganesan([rganesa@calstatela.edu](mailto:rganesa@calstatela.edu))

Manali Joshi([mjoshi11@calstatela.edu](mailto:mjoshi11@calstatela.edu)), Mittal Vaghela([mvaghel@calstatela.edu](mailto:mvaghel@calstatela.edu))

05/18/2017

**Predictive Analysis of Income**

**Objectives**

The aim of this Tutorial is to predict the Income and Department of an employee based on the available features form the dataset by utilizing Machine Learning Algorithms and build accurate models using AzureML:

* Get Three data sets from the links
* Create Azure ML models
* Use appropriate Machine Learning algorithms:

1. **Regression**: Boosted Decision Tree Regression and Decision Forest Regression
2. **Classification**: Multiclass Logistic Regression and Multiclass Neural Network

* Use Python scripts
* Choose best model based on appropriate metrics- RMSE, Accuracy, Precision, Recall.

**What You’ll Need**

To complete this lab, you will need the following:

• An Azure ML account

• A web browser and Internet connection

• The lab files for this lab

• Python Anaconda – our classroom should have a python editor

• The **LA Income Data, SF Income Data and NY Salary Data** datasets (see *Prepare the Data* steps below)

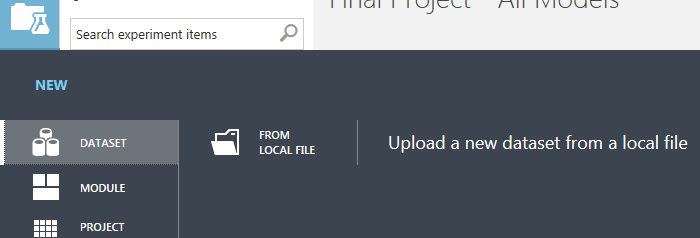
Step 1: Prepare and Upload Data

1. Open a browser and browse to <https://studio.azureml.net> .Then sign in using the Microsoft account associated with your Azure ML account.
2. Download the data from the Below Three Links:

* <https://data.lacounty.gov/Operations/LA-County-Employee-Salaries/8rdv-6nb6>
* <https://catalog.data.gov/dataset/employee-compensation-53987>
* <https://data.ny.gov/Transparency/Salary-Information-for-State-Authorities/unag-2p27>

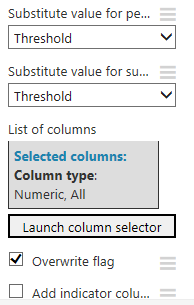
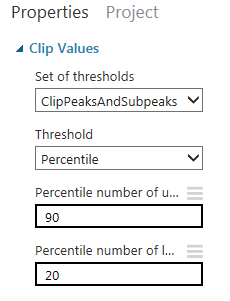
1. Save the data on your computer as CSV and name the three as LA\_County\_Employee\_Salaries, New York Wage and seatle wage.
2. Once downloaded, Open the files in Excel and Select all the Columns containing Salary Figures with a “$” sign. To remove the $ symbol, Right click🡪 Format cells🡪Change Format type to Number.
3. At Azure ML studio you need to upload the input csv files as follows:

New🡪Datasets🡪From Local File

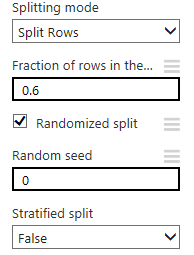


Step 2: Model Building for LA Employee Salary Predictions

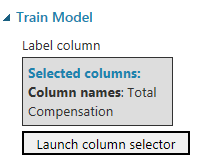
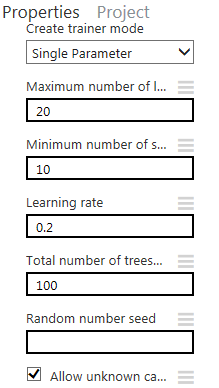
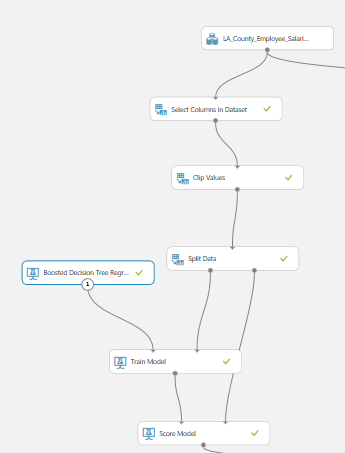
1. Create an Experiment and name it Final Project - All Models. Go to My Datasets from the Saved Datasets and search for the LA\_County\_Employee\_Salaries dataset you uploaded. Drag it on the Canvas.
2. Next, drag the Select Columns in Dataset module to the canvas and click Launch column selector and select the following columns **Total Compensation, Pension Contributions, LTD Life & Medical Tax**.
3. Thereafter, drag the Clip Values module onto the canvas and set the following parameters for the same.



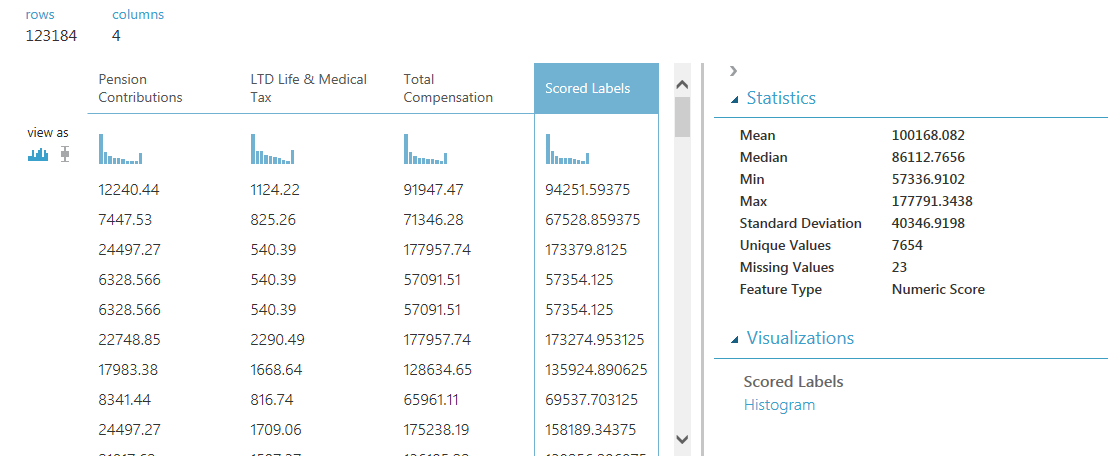
1. Choose the Split Data module and drag it to the canvas and set the following Parameters:



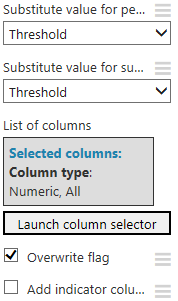
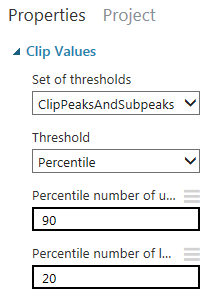
1. Going ahead, choose the **Boosted Decision Tree Regression** Algorithm, Train Model and Test Model onto the canvas and make All the parameters and connections as shown below:



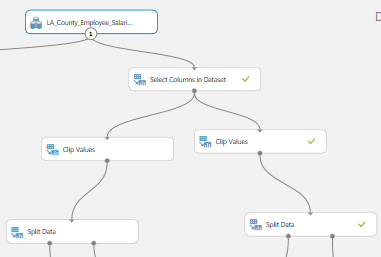
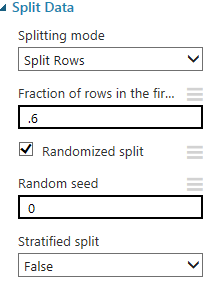
1. Right click and Visualize the Scored dataset output port. The Scored Labels are the predicted values of our test. You should see a similar result as below:



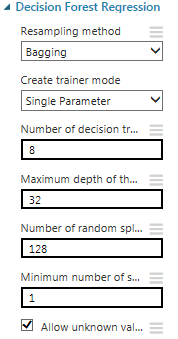
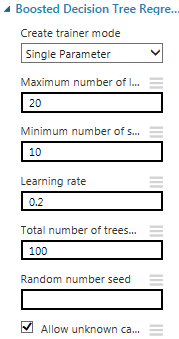
1. Going ahead, Drag the second Select Columns in Dataset module onto the canvas besides the first one and select the **Medical, Dental, Vision, Total Compensation, LTD Life & Medical Tax, Pension Contributions** columns using the Launch column selector.
2. Also, drag two more Clip Values module and make the connections as shown in the figure at step 9. Make the relevant parameters for both as below:



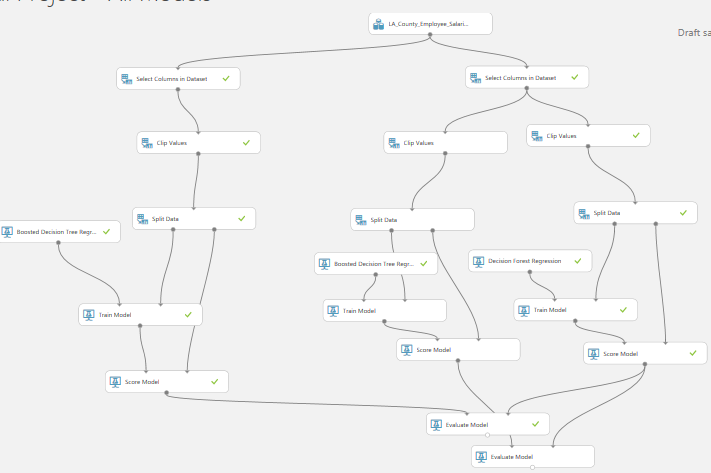
1. Drag two more Split Data modules below the Clip Values and make the connections as per the below figure. Set the parameters of both the Split Data module as below:



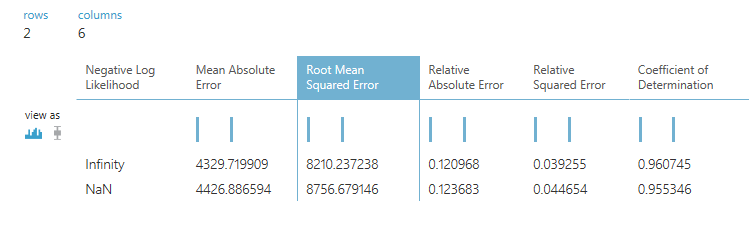
1. Drag the **Boosted Decision Tree Regression** and the **Decision Forest Regression** modules below the second and the third Split Data Modules respectively and set the following parameters for them:



1. Thereafter drag two more Train Model, Score Model and Evaluate Model blocks onto the canvas. For the Train Model, Select the Total Compensation columns for both. Make all the connections as shown below. Your final Canvas should look exactly like this.



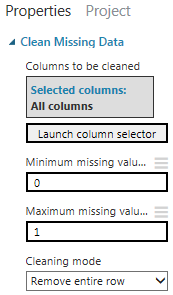
1. At the end, Save and Run the Experiment. Visualize the output of the bottom most Evaluate model and you can see that the **RMSE** for the second **Boosted Decision Tree Regression** model with an extra feature was the lowest at **8210.23** and had a **Coefficient of Determination** of **0.96** which is better than that of the Decision Forest Regression Algorithm. Hence, we can conclude that the BDTR algorithm correctly predicts the Income for the Los Angeles County.



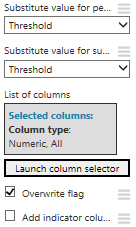
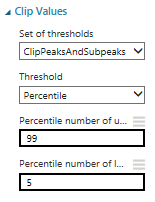
Step 3: Model Building for New York Income Predictions

**Objective:** To predict the Department of an Employee in New York based on his Base Annualized Earnings by using Classification Algorithms.

* 1. Go to My Datasets from the Saved Datasets and search for the New York Wage dataset you uploaded. Drag it on the Canvas.
  2. Then, Drag the Select Columns in Dataset module, make connections with dataset and select the **Base Annualized Salary, Group, Title** columns from the launch columns selector.
  3. Drag the Clean Missing Module and connect the Result Dataset output from select columns module to the input of Clean Missing data. Set the following properties for it:



* 1. Thereafter drag the Clip Values Module to remove the outliers from our data. Set its properties as follows:



* 1. Now we will drag the Execute Python script block onto the canvas. Here, we will feed the Python script for converting the Base Annualized Earnings into categorical data by making suitable bins. Connect the output of Clip Values to the Dataset1 input port of Execute Python Script. Copy and paste the below code into the Python Editor available in the Execute Python Script module:

def azureml\_main(frame1):

## Quantize the wind into categories using cut with explicit columns.

import pandas as pd

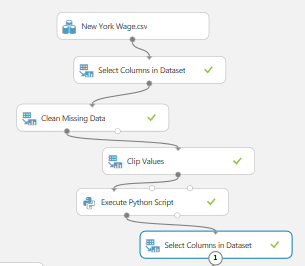
bins = [0, 10000, 20000, 30000, 40000,50000,60000,70000,80000,90000,100000,110000,130000,140000,150000,160000,170000,180000,190000,200000,210000,220000,230000,240000,250000,260000,270000,280000,290000,300000]

frame1['base\_sal'] = pd.cut(frame1['Base Annualized Salary'], bins)

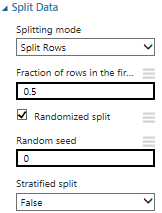
#frame1['base\_compe'] = pd.cut(frame1['Total Earnings'],bins)

return frame1

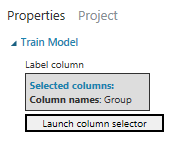
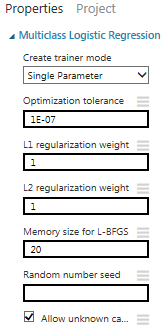
* 1. Moving forward, drag one more Select Columns module and connect the Result Dataset output as its input. Then through the Launch Column Selector choose only **Group, Title, base\_sal** columns. Your Canvas should look as follows up to this point:



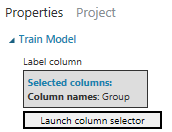
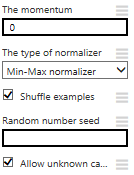
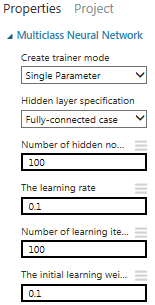
* 1. Drag the Split Data module below the Select Columns and connect them. Set the following properties:



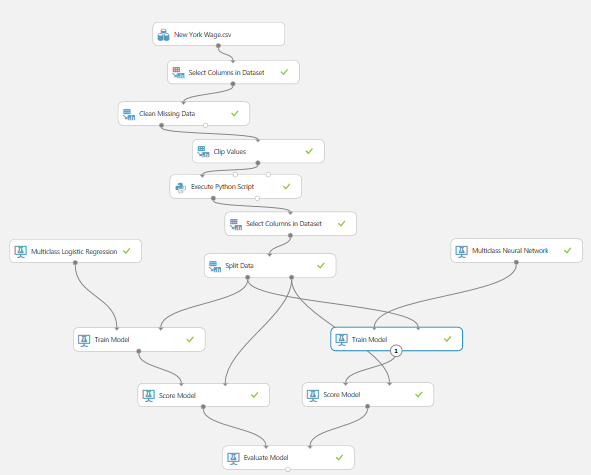
* 1. Next, drag **Multiclass Logistic Regression** algorithm and the Train Model module. Connect the Untrained model output of MLR to the untrained model input of Train model. Also, connect the Results Dataset1 output of the Split data to the Dataset input of Train Model. Set the properties for both as shown below:



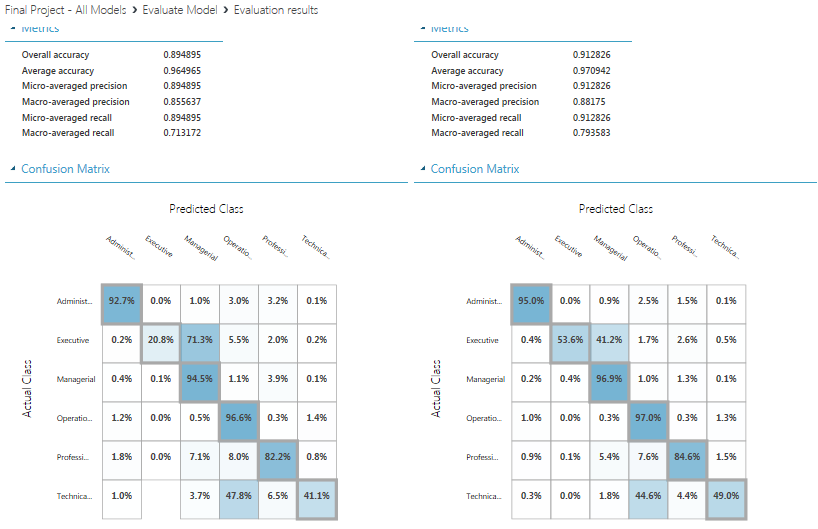
* 1. Drag one more Train Model module and **Multiclass Neural Network** algorithm onto the canvas. Also drag two Score Model module and one Evaluate Model module and make the connections as shown in step 10. Set the properties for Multiclass Neural Networks and the second Trian Model as follows:



* 1. Finally, your canvas should look exactly as shown below. Check all the connections and save and Run the Experiment.



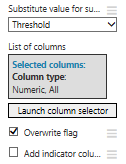
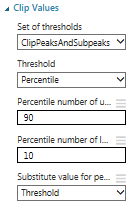
* 1. At the last, Visualize the output of the Evaluate Model. Here we can see that the Multiclass Neural Network Algorithm performs better than the Multiclass Logistic Regression as it has a better **Overall Accuracy of 91.2%** and predicts the Department based on Salary with a good accuracy. Also, it has good precision and recall values. Also, the Confusion Matrix looks more convincing for **Multiclass Neural network**.



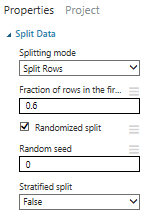
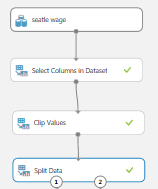
Step 4: Model Building for San Francisco Salaries

**Objective:** To predict the Salary of an Employee in San Francisco based on Other Benefits by using Regression Algorithms.

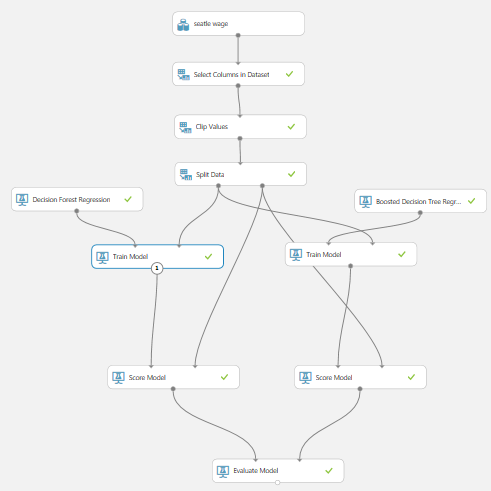
* 1. Go to My Datasets and drag the San Francisco Dataset named seatle wage onto the canvas. Also, drag the Select Columns module, make connections and select the **Retirement, Salaries, Health/Dental and Total Compensation** columns using the Launch Column Selector option.
  2. Then drag the Clip Values module to remove the outliers from the data. Set the following Parameters after making the connection with select column module.



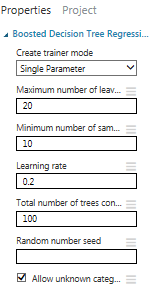
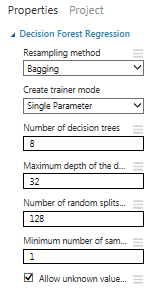
1. Next, we need to split the data to Train and Test, hence drag the Split Data module onto the canvas and do the connections as shown below. Also set the following properties:



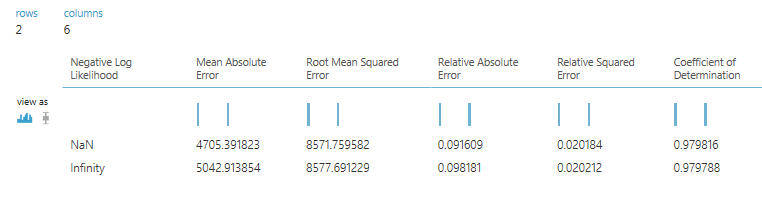
1. Thereafter drag the **Decision Forest Regression** and **Boosted Decision Tree Regression** algorithms. Also, drag two Train Model, two Score Model and one Evaluate Model modules onto the canvas and connect them as shown in the below figure.



1. In the two Train Models, select the **Total Compensation** column from the Launch Column Selector option. Moreover, make the following changes in the Properties of the two algorithms used:



1. Lastly, Save and Run the experiment. Then Visualize the output of the Evaluate model. Here, we can see that there only marginal difference in the RMSE values and the Coefficient of Determination. The Decision Forest Regression (DFR) model has an **RMSE** of **8571.75** marginally less than that of Boosted Decision Tree Regression (BDTR). But, the **COD value is marginally better for the BDTR**. Hence, we can say that either of the Algorithms work well, but if RMSE is to be considered, the **Decision Forest Regression** model performs well for predicting the Incomes for the **San Francisco dataset**.



References

1. **URL’s of Data Sources:**

* <https://data.lacounty.gov/Operations/LA-County-Employee-Salaries/8rdv-6nb6>
* <https://catalog.data.gov/dataset/employee-compensation-53987>
* <https://data.ny.gov/Transparency/Salary-Information-for-State-Authorities/unag-2p27>

1. **GitHub URL:** <https://github.com/roshik2016/CIS5560>
2. **Other Reference:** <https://studio.azureml.net>