4. Data Mining with Azure ML Studio

4.1 Getting Started with Azure Machine Learning Studio

In this section we will be familiarizing ourselves with Azure Machine Learning ("ML") Studio. We will create a dedicated storage account for our experiment and a workspace within our account, learn how to access the workspace from the Azure Portal, and finally create our first experiment.
In order to get started and begin your first exercise with Azure ML, you must sign up for a free trial. You can register at: `http://azure.microsoft.com/en-us/pricing/free-trial/`

4.1.1 Exercise: Creating an Azure Machine Learning Studio Workspace

Once you have a dedicated Azure storage account, you can create an Azure ML Studio workspace.

1. Create a new Azure ML Studio workspace by selecting:
   +New > **Data Services** > **Machine Learning** > **Quick Create** (Figure: 4.1, 4.2) .

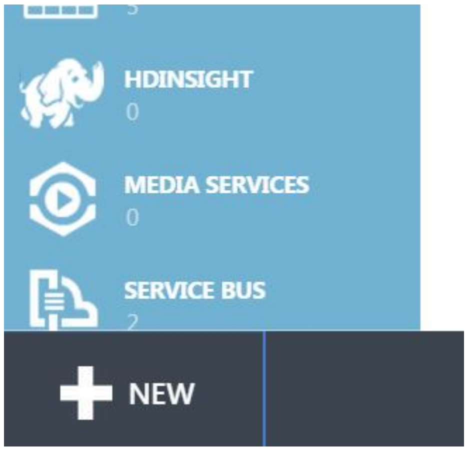
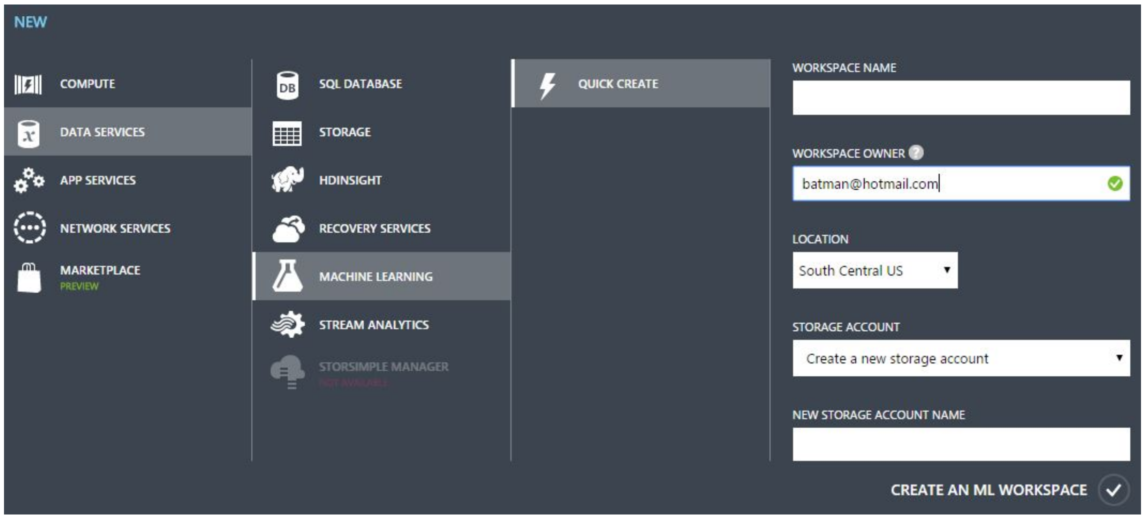


Figure 4.1: Create a new workspace



WORKSPACE NAME

WORKSPACE OWNER: batman@hotmail.com

LOCATION: South Central US

STORAGE ACCOUNT: Create a new storage account

NEW STORAGE ACCOUNT NAME

Figure 4.2: Create a new workspace

2. In the **Workspace Name** box, assign a globally unique name.
3. In the **Workspace Owner** box, input the administrative email for your Azure account, preferrably a hotmail account.
4. In the **Storage Account** dropdown, select "Create a new storage account".
5. In the **New Storage Account Name**, give your blob storage a globally unique name.

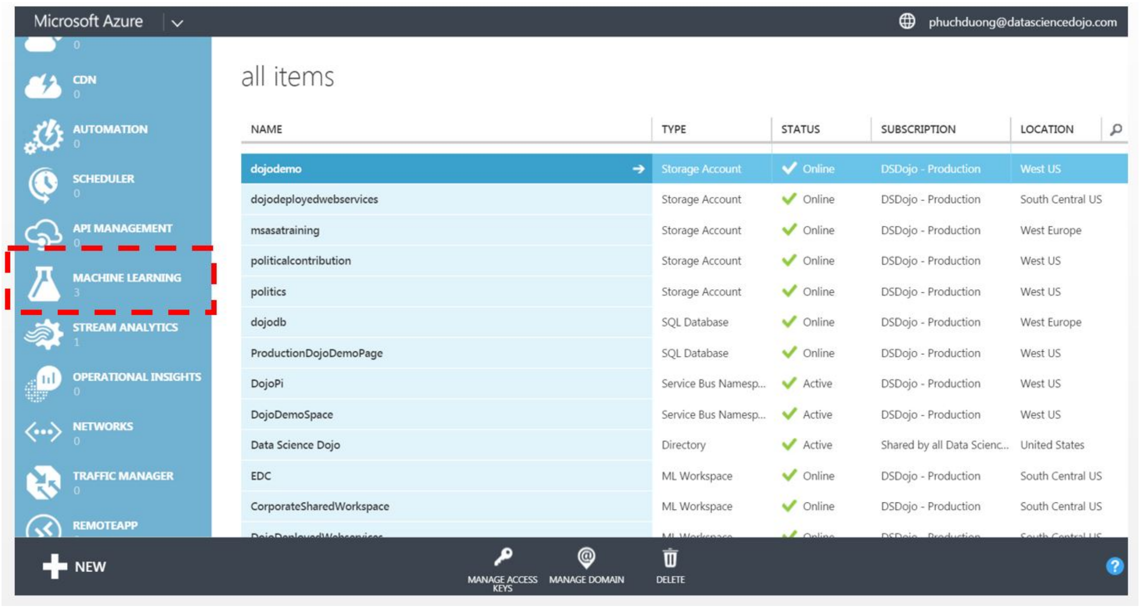
Click the check mark once the credentials have been populated to send off a workspace request to Azure. The workspace will take at least two minutes to setup. Accidently deleting the blob storage associated with your Azure ML workspace will corrupt the workspace and render it unusable.

Tip You can invite others to collaborate in your workspace by adding them as users to the account under **Settings**. You can also copy and paste experiments across workspaces.

4.1.2 Exercise: Accessing your Azure Machine Learning Workspace

You may now access your Azure ML workspace.

1. Within the Azure Portal, select **Machine Learning** (Figure: 4.3).



all items

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
dojodemo	Storage Account	✓ Online	DSDojo - Production	West US
dojodeployedwebservices	Storage Account	✓ Online	DSDojo - Production	South Central US
msasatraining	Storage Account	✓ Online	DSDojo - Production	West Europe
politicalcontribution	Storage Account	✓ Online	DSDojo - Production	West US
politics	Storage Account	✓ Online	DSDojo - Production	West US
dojodb	SQL Database	✓ Online	DSDojo - Production	West Europe
ProductionDojoDemoPage	SQL Database	✓ Online	DSDojo - Production	West US
DojoPi	Service Bus Namesp...	✓ Active	DSDojo - Production	West US
DojoDemoSpace	Service Bus Namesp...	✓ Active	DSDojo - Production	West US
Data Science Dojo	Directory	✓ Active	Shared by all Data Scienc...	United States
EDC	ML Workspace	✓ Online	DSDojo - Production	South Central US
CorporateSharedWorkspace	ML Workspace	✓ Online	DSDojo - Production	South Central US

Figure 4.3: Access your machine learning workspace

2. Select the workspace that you just created in Exercise: Creating an Azure Machine Learning Studio Workspace.
3. Select "Access your Workspace" (Figure: 4.4) A new window will appear.

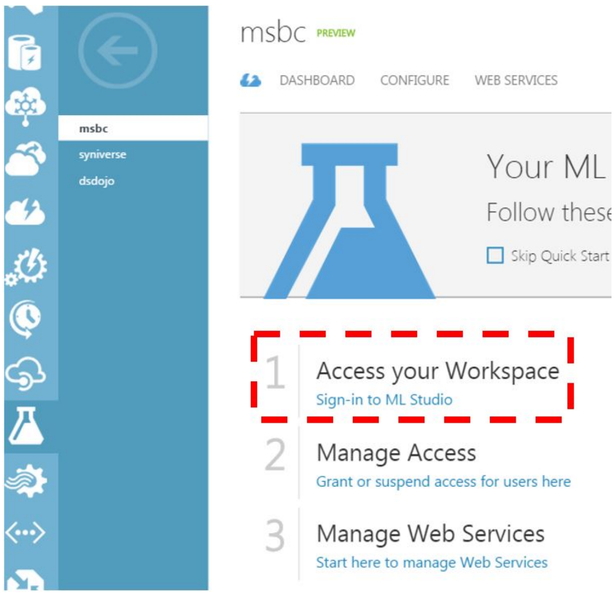


Figure 4.4: Access your workspace

4.1.3 Exercise: Creating your First Experiment

Data Science is an interdisciplinary art and science. It borrows terms from other disciplines, especially the sciences. In this tradition, a project in data science is called an experiment.

1. To create a new experiment, select **+New** > **Experiment** > "Blank Experiment" (Figure: 4.5) .

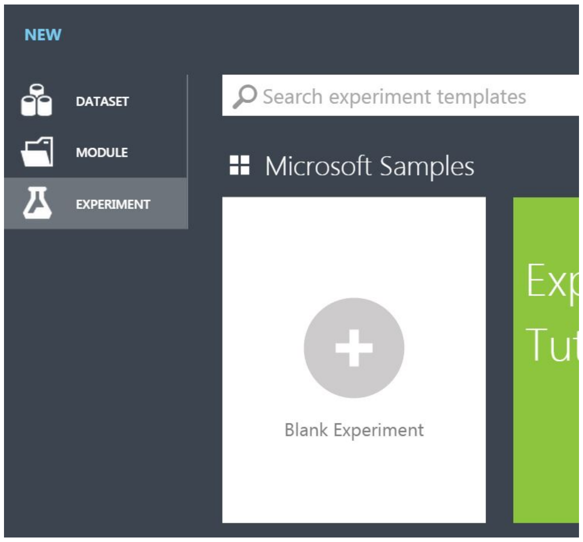


Figure 4.5: Create a new experiment

2. Name your experiment in the "Experiment Name" field.

We aren't ready to save our experiment yet, so for now we will move on.

4.2 Methods of Ingress and Egress with Azure Machine Learning Studio

4.2.1 Exercise: Reading a Dataset from a Local File

The first dataset we will be using is the go-to database when getting started with data science. The data describes features of an iris plant in an attempt to predict its class.
To retrieve the dataset, Google "UCI Iris Data" or go to:
`http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data`
Notice how commas differentiate each value. This allows us to know that the elements can be read as comma separated values ("CSV"). Excel files and delimited text files can be read as CSV as well. Also notice that the data does not have headers. The model will eventually require headers but we will define these later on.

1. Download and save the text as a CSV file. For example "filename.csv".
2. In Azure ML Studio, select **+New** > **Dataset** > **From Local File**.
3. Please note that by default, Azure ML ships with a dataset called "Iris Two Class Data". To avoid confusion, give your dataset a unique name, then import.
4. To verify that your data has been imported, go into any experiment and look under the directory **Saved Datasets** (Figure: 4.6). You should see the name you chose for your data listed.

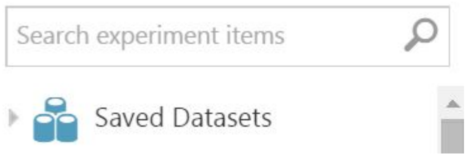


Figure 4.6: Saved dataset directory

5. Now that your experiment has a module in it, you can now save your experiment. Select "Save As" on the menu at the bottom of your screen (Figure: 4.7).

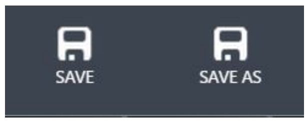


Figure 4.7: Save the experiment

4.2.2 Exercise: Reading a Dataset from a URL

1. To begin, use the search bar to find the **Reader** module within your experiment. Drag and drop the module from the menu on the left (Figure: 4.8).

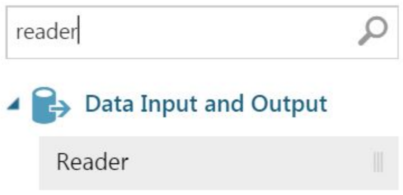
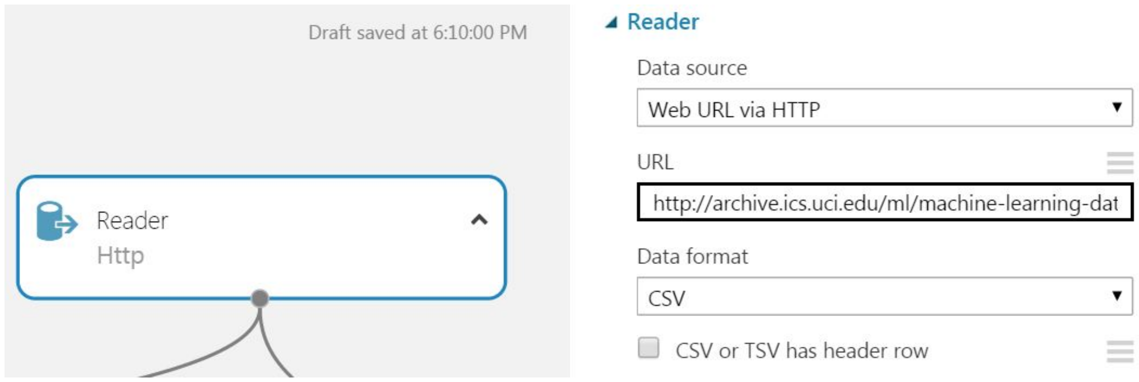


Figure 4.8: Search for the Reader module

2. In the**Reader** settings for "Please specify data source" select "Http".
3. In the "URL" box, enter the URL of the iris data set:
`http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data`.
4. In the "Date format" drop down, select "CSV".
5. Leave "CSV or TSV has header row" unchecked (Figure: 4.9).



Reader

Data source
Web URL via HTTP

URL
http://archive.ics.uci.edu/ml/machine-learning-dat

Data format
CSV

☐ CSV or TSV has header row

Figure 4.9: Reader module settings

6. Select **Run** to import and parse the experiment (Figure: 4.10).

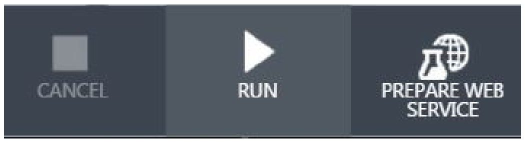


Figure 4.10: Run the experiment

7. In order to preserve the dataset, we must save our work. Save the output of your experiment by right-clicking the bottom middle node of the **Reader** module again. Select "Save as Dataset". Please note that by default, Azure ML ships with a dataset called "Iris Two Class Data". To avoid confusion, give your dataset a unique name, then import.
8. To verify that your dataset has been successfully imported, go into any experiment and look under the directory **Saved Datasets**. You should see the name you chose for your data listed.

4.2.3 Exercise: Reading a Dataset from Azure Blob Storage

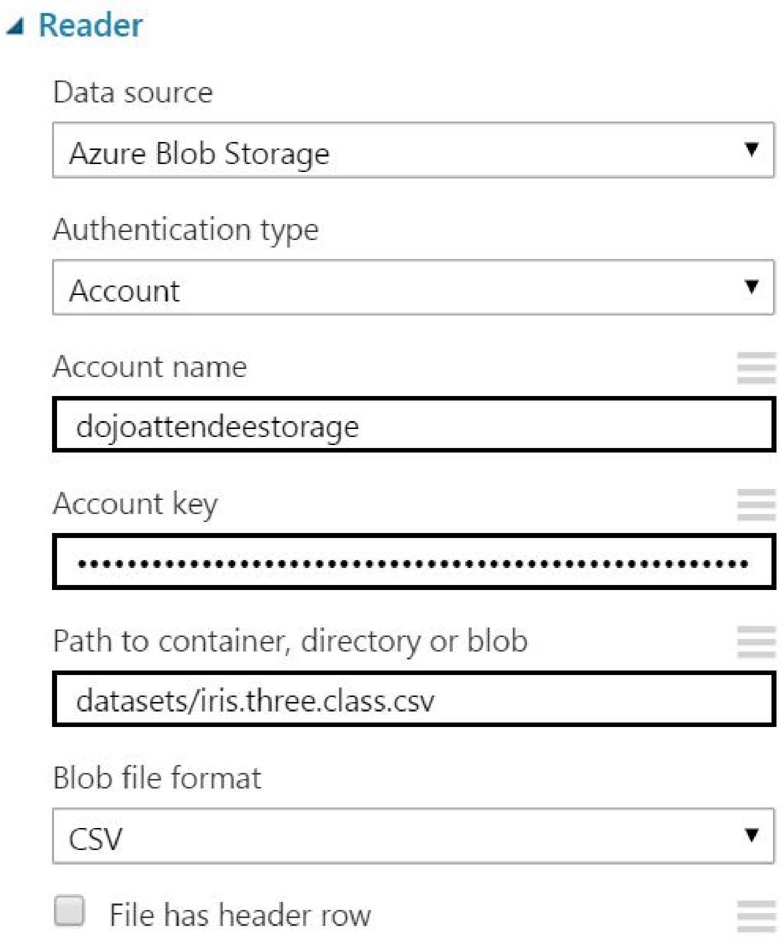
1. To begin, use the search bar to find the **Reader** module within your experiment. Drag and drop the module from the menu on the left.
2. For the required fields, input the information from Table: 4.1.

Required Field	Input
Data source	Azure Blob Storage
Authentication type	Account
Account name	dojoattendeestorage
Account key	aKQOxU3As1BsS3yT2bh HkJ/icCICJPpL1tdWKxQ+tP BNk6DbykV4qd3HGlFPZ N/3TdiUHuM/Quk 9DPUeQu7M8A==
Path to container, directory or blob	datasets/iris.three.class.csv
Blob file format	CSV
File has header row	Unchecked

Table 4.1: Azure Blob Storage Log-In Details

Tip Note that "dojoattendeestorage" is the container. Containers contain blobs which are essentially files in the Azure Cloud itself. For those who are familiar with web development, this is equivalent to an FTP.

Figure: 4.11 depicts a sample of what your **Reader** module will look like after all of the above steps have been followed.


Reader

Data source

Azure Blob Storage

Authentication type

Account

Account name

dojoattendeestorage

Account key

••

Path to container, directory or blob

datasets/iris.three.class.csv

Blob file format

CSV

☐ File has header row

Figure 4.11: Reader module settings

3. Select **Run** to import and parse the experiment.
4. In order to preserve the dataset, we must save our work. Save the output of your experiment by right-clicking the bottom middle node of the **Reader** module again. Select "Save as Dataset". Please note that by default, Azure ML ships with a dataset called "Iris Two Class Data". To avoid confusion, give your dataset a unique name, then import.
5. To verify that your dataset has been successfully imported, go into any experiment and look under the directory **Saved Datasets**. You should see the name you chose for your data listed.

4.2.4 Exercise: Writing a Dataset to Azure Blob Storage

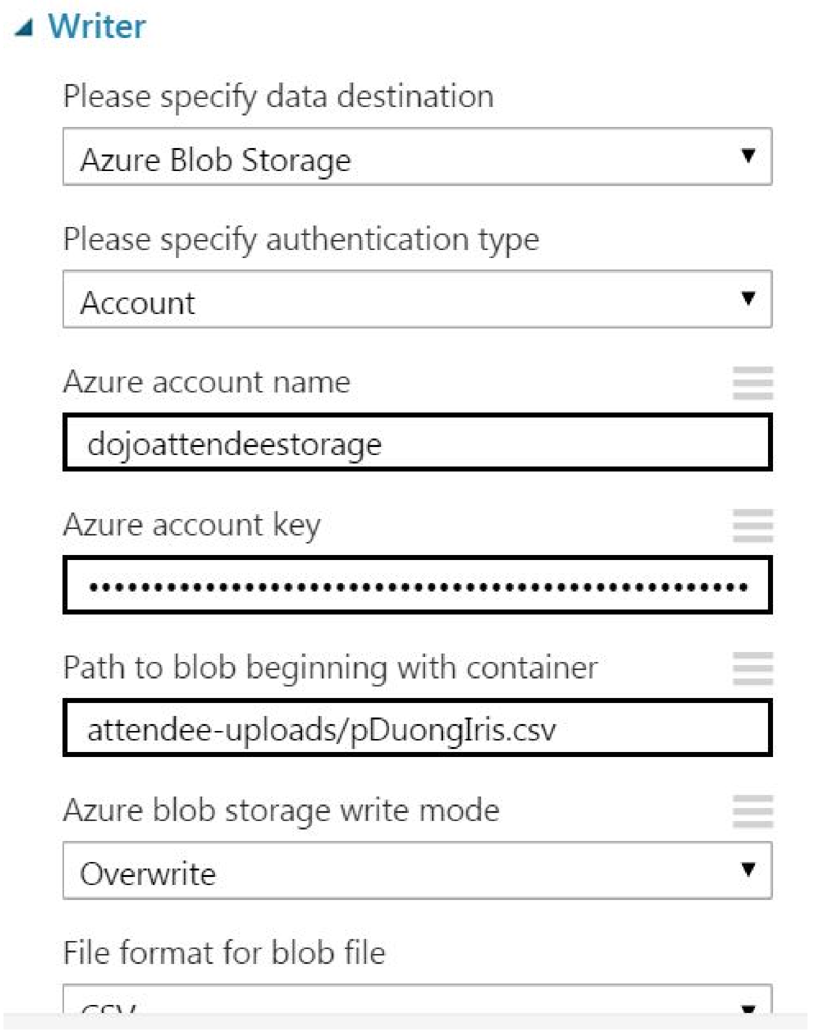
1. Go into the directory **Saved Datasets** and drag any dataset into your workspace.
2. Search for the **Writer** module in the search box. Drag the module into your workspace and connect it to your dataset.
3. For the required fields, input the information from (Table: 4.2).

Required Field	Input
Please specify data destination	Azure Blob Storage
Please specify authentication type	Account
Azure account name	dojoattendeestorage
Azure account key	aKQOxU3As1BsS3yT2b hHkJ/icCICJPpL1tdWKxQ+tPB Nk6DbykV4qd3HGlFPZN/3 TdiUHuM/Quk9DPUeQu7M8A==
Path to blob beginning with container	attendee-uploads/<file-name>.csv
Azure blob storage write mode	Overwrite
Azure blob storage write mode	CSV
Write blob header row	Unchecked

Table 4.2: Azure Blob Storage Log-In Details

Tip Normally when prompted for the "Path to blob beginning with container", you can choose any file name you would like. However, since many people will be writing to this blob during this exercise, do not name the file iris.csv. Name the file with your first initial, last name, then iris as one word (i.e. John Smith or jSmithiris.csv).

Figure: 4.12 depicts a sample of what your **Writer** module will look like after all of the above steps have been followed.


Writer

Please specify data destination: Azure Blob Storage

Please specify authentication type: Account

Azure account name: dojoattendeestorage

Azure account key: ••

Path to blob beginning with container: attendee-uploads/pDuongIris.csv

Azure blob storage write mode: Overwrite

File format for blob file

Figure 4.12: Sample Writer module settings

4.3 Visualizing, Exploring, Cleaning, and Manipulating Data

4.3.1 About the Data

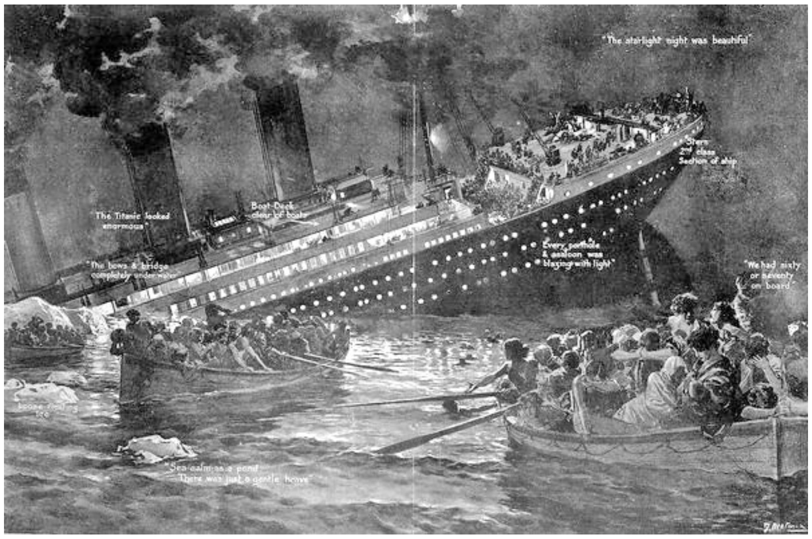


Figure 4.13: Illustration of the sinking of the Titanic. Source: National Maritime Museum

For most of this chapter we will be working with the Titanic dataset. The Titanic data is a good beginner's dataset to start learning how to data mine. The sinking of the RMS Titanic occurred in 1912 and is one of the most infamous shipwrecks in history. 1,502 out of 2,224 passengers were killed in the tragedy and the incident caused an international backlash for ship safety reform. Although the mass loss of life is mainly attributed to the lack of life vessels and other elements of chance, some groups were more likely to survive than others. We will use the power of machine learning to uncover which types of individuals were more likely to survive. To begin, we will first procure the dataset.

4.3.2 Exercise: Obtaining the Titanic sample data

You will be reading in the dataset from our online GitHub repository by using a **Reader** module.

1. Drag and drop a **Reader** module from the menu on left (Figure: 4.14).

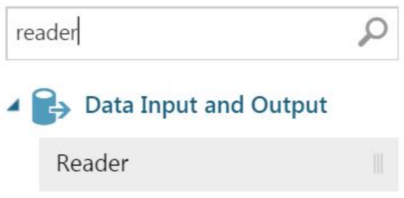


Figure 4.14: Search for the Reader module

2. Set the **Reader** module with the settings found in 4.3.

Field	Setting and Inputs
Data Source	Web URL via HTTP
URL	`https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv`
Data format	CSV
CSV or TSV has header row	Checked

Table 4.3: Reader module settings

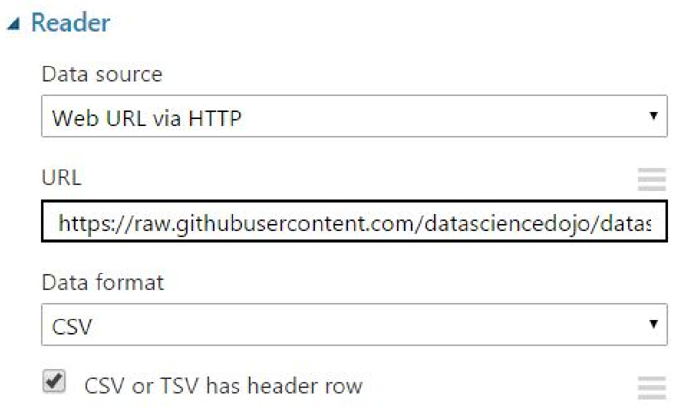
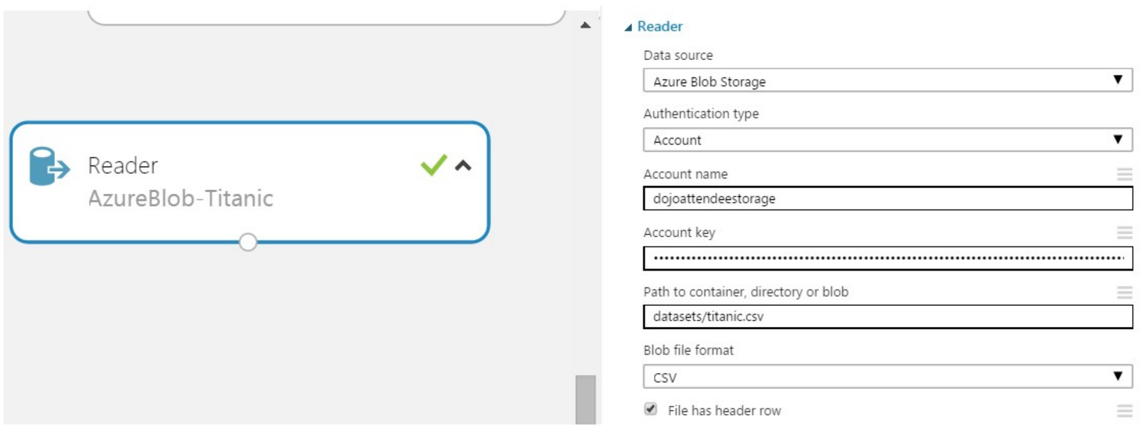

Reader
Data source: Web URL via HTTP
URL: https://raw.githubusercontent.com/datasciencedojo/datas
Data format: CSV
☑ CSV or TSV has header row

Figure 4.15: Reader module settings

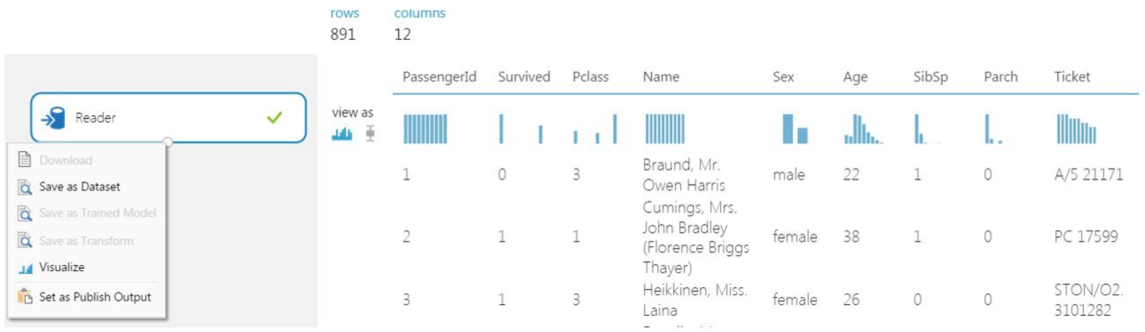
Figure: 4.16 depicts a sample of what your **Reader** module will look like after all of the above steps have been followed.



Reader
Data source: Azure Blob Storage
Authentication type: Account
Account name: dojoattendeestorage
Account key: ••
Path to container, directory or blob: datasets/titanic.csv
Blob file format: CSV
☑ File has header row

Figure 4.16: Reader module settings

3. Select **Run** to execute the import and parse.
4. In order to preserve the dataset, we must save our work. Save the output of your experiment by right-clicking the bottom middle node of the **Reader** module again. Select "Save as Dataset".



rows 891 columns 12

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282

Figure 4.17: Visualize the Reader module

5. To verify that your dataset has been successfully imported, go into any experiment and look under the directory **Saved Datasets**. You should see the name you chose for your data listed.

4.3.3 Titanic Dataset Key

The Titanic dataset is made up of qualitative and quantitative information. The Titanic key describes the meaning variables and their corresponding values. The key can also be found at: `https://www.kaggle.com/c/titanic/data`.

Variable Descriptions

Column Name	Meaning	Notes
survival	Survival	(0 = No; 1 = Yes)
pclass	Passenger Class	(1 = 1st; 2 = 2nd; 3 = 3rd)
name	Name	
sex	Gender	
age	Age	
sibsp	Number of Siblings/Spouses Aboard	
parch	Number of Parents/Children Aboard	
ticket	Ticket Number	
fare	Passenger Fare	In 1910 USD
cabin	Cabin	
embarked	Port of Embarkation	(C = Cherbourg; Q = Queenstown; S = Southampton)

Special Notes

- Pclass is a way to infer socio-economic status (SES)
  1st Upper; 2nd Middle; 3rd Lower
- Age is in Years; age is fractional if the passenger age is less than one
  If the age is Estimated it is in the form "xx.5"
- With respect to the family relation variables (i.e. sibsp and parch) some relations were ignored. The following are the definitions used for sibsp and parch.
  – Sibling: Brother, sister, stepbrother, or stepsister of the passenger
  – Spouse: Husband or wife of the passenger (mistresses and fiances Ignored)
  – Parent: Mother or father of the passenger
  – Child: Son, daughter, stepson, or stepdaughter of the passenger

- Other family relatives excluded from this study include cousins, nephews, nieces, aunts, uncles, and in-laws. Some children travelled only with a nanny, therefore parch=0 for them. In addition, some passengers travelled with close friends or neighbors in a village, however, the definitions do not support such relations.

4.3.4 Exercise: Casting Columns

Although the Titanic dataset contains categorical data types, by default Azure will treat them as sequential numbers. Therefore we must tell Azure which columns are categorical.

1. Go into **Saved Datasets** and find the Titanic dataset you just obtained. Drag the dataset into your experiment's workspace (Figure: 4.18)

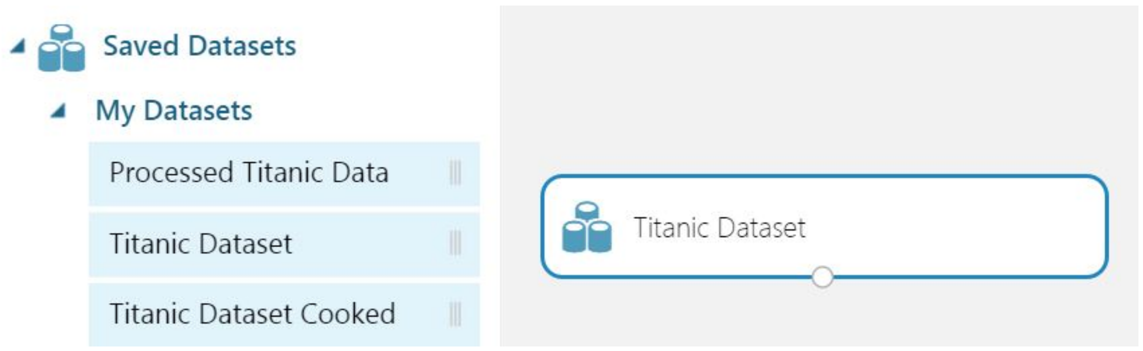


Figure 4.18: Drag the Titanic Dataset into the workspace

2. Right-click on the bottom center node of the dataset. Select "Visualize" to see the output. Verify that it looks like Figure: 4.20.

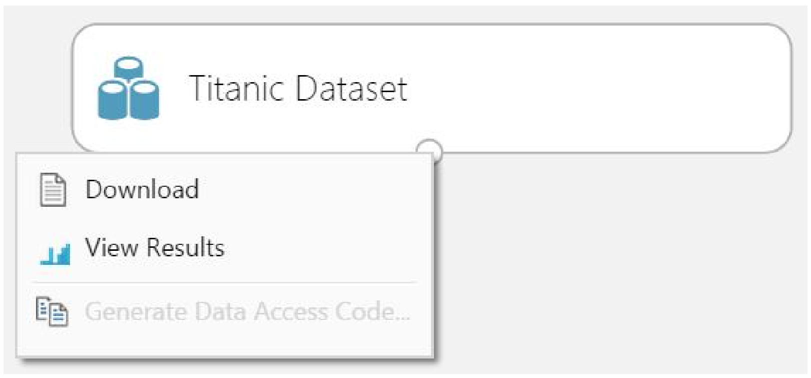
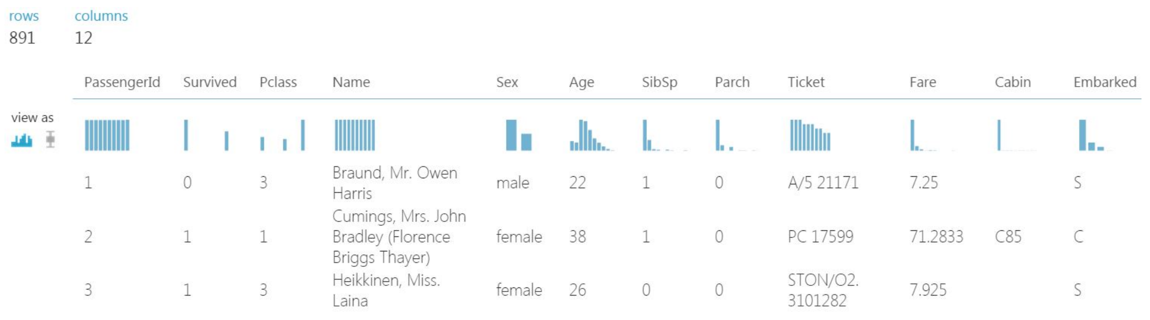


Figure 4.19: Visualize the Titanic Dataset

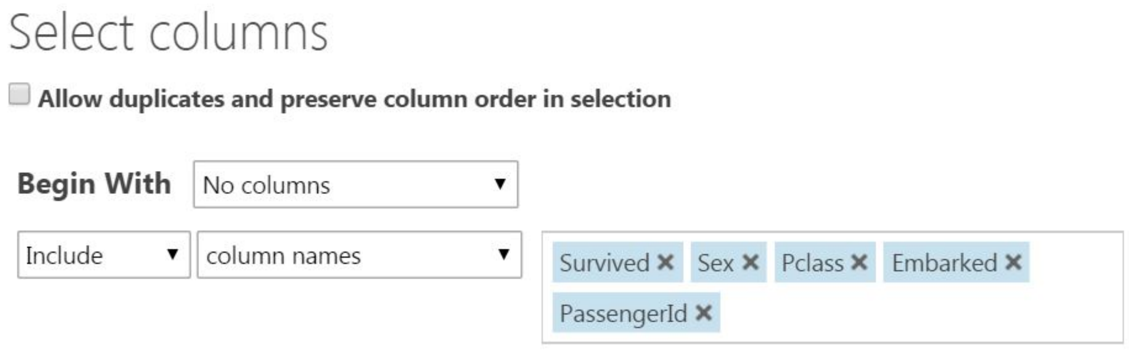

rows 891 columns 12

view as

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S

Figure 4.20: Visualize the Titanic Dataset

3. After verifying the output, we will cast categorical values to the corresponding columns. To begin, search for the **Metadata Editor** in the left menu and drag the module into the workspace.
4. Connect the **Metadata Editor** to the dataset and launch the "column selector" within the editor.
5. Select "Launch column selector". For the box of chosen values, add "Survived", "Sex", "Pclass", "Embarked", and "PassengerId". Leave all of the other fields unchanged (Figure: 4.21).


Select columns

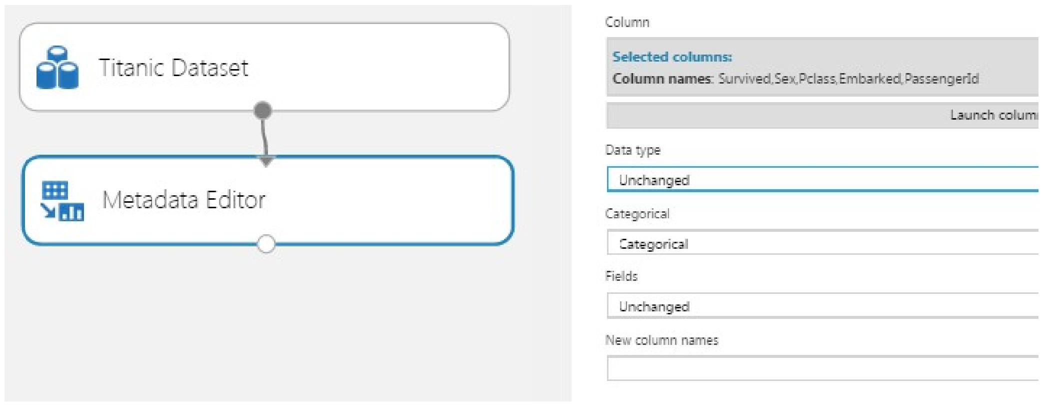
Allow duplicates and preserve column order in selection

Begin With: No columns

Include | column names | Survived, Sex, Pclass, Embarked, PassengerId

Figure 4.21: Column selector settings

6. After applying your settings in the "column selector", change the "Categorical" field to "Make Categorical" in the **Metadata Editor**. Keep all other fields as they are (Figure: 4.22).



Column
Selected columns:
Column names: Survived,Sex,Pclass,Embarked,PassengerId

Data type: Unchanged
Categorical: Categorical
Fields: Unchanged
New column names:

Figure 4.22: Metadata editor settings

7. It is a good habit to label the different modules so you can remember their function as more modules get added to the workspace. To do this, double click on the module and type the label. For this **Metadata Editor** label it “Categorical Casting”.

4.3.5 Exercise: Data Visualization and Exploration

1. One way to view the distribution of your data is with a histogram. To do this, right-click on the **Metadata Editor** and select “Visualize”. Select the “Survived” column and make sure the “view as” is set to the picture of a histogram (Figure: 4.23)

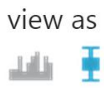


Figure 4.23: Histogram icon

2. A menu will populate the right hand side of the screen. Select the “Visualizations” dropdown. A histogram like that in Figure: 4.24 will now appear in the window.

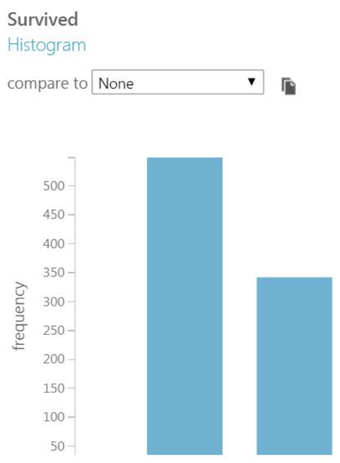


Figure 4.24: Survived histogram

How is the data distributed in terms of survived and deceased passengers? Did more people survive or perish? (Figure: 4.25).

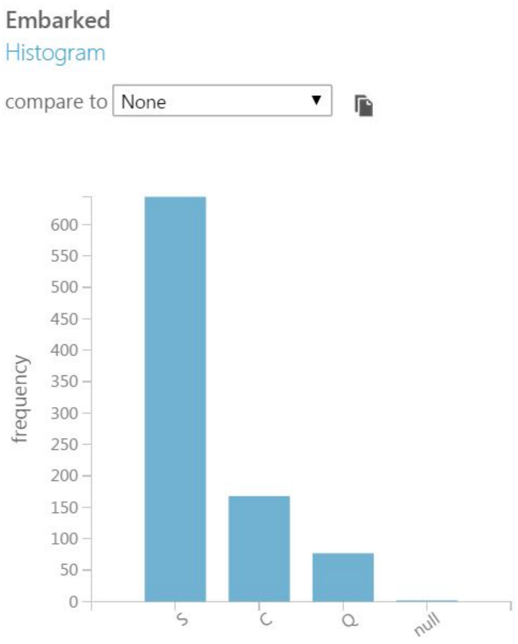
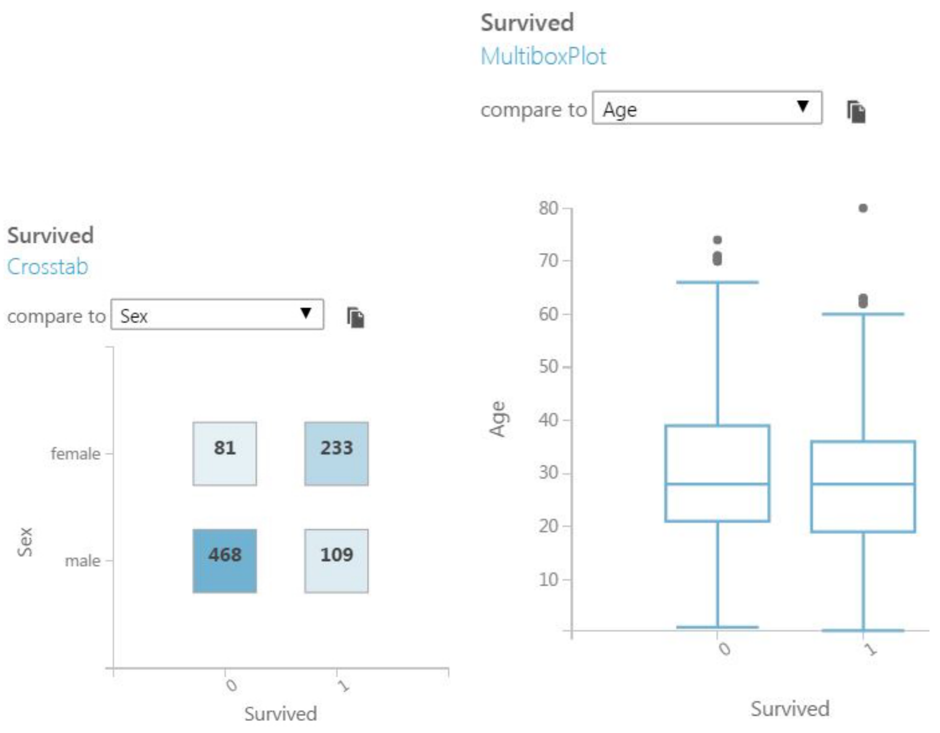


Figure 4.25: Embarked histogram

3. Select the "Embarked" column from the left hand side. What do you notice about this histogram? Which port of embarkation did most of the passengers come from? The least?
4. It's possible to compare different variables by comparing visualizations. Let's begin with comparing gender and survival. Select the "Survival" column again. On the top of the histogram under "compare to" select "Sex" (Figure: 4.26)
   Based on the graph, did gender play a part in the chances of survival?
5. Change "compare to" to "age" (Figure: 4.26)
   Is there a relationship between age and survival? What was the oldest age of survival? The oldest age of death?
6. Select the "Sex" column from the left hand side. Set the "compare to" dropdown to "Age". What was the distribution of males and females among age groups? Which group had an older distribution of passengers?
7. Set the "compare to" dropdown to "Fare". Make sure "log scale" is checked.
   What is the relationship between gender and ticket price? Which category generally paid more?



compare to Sex

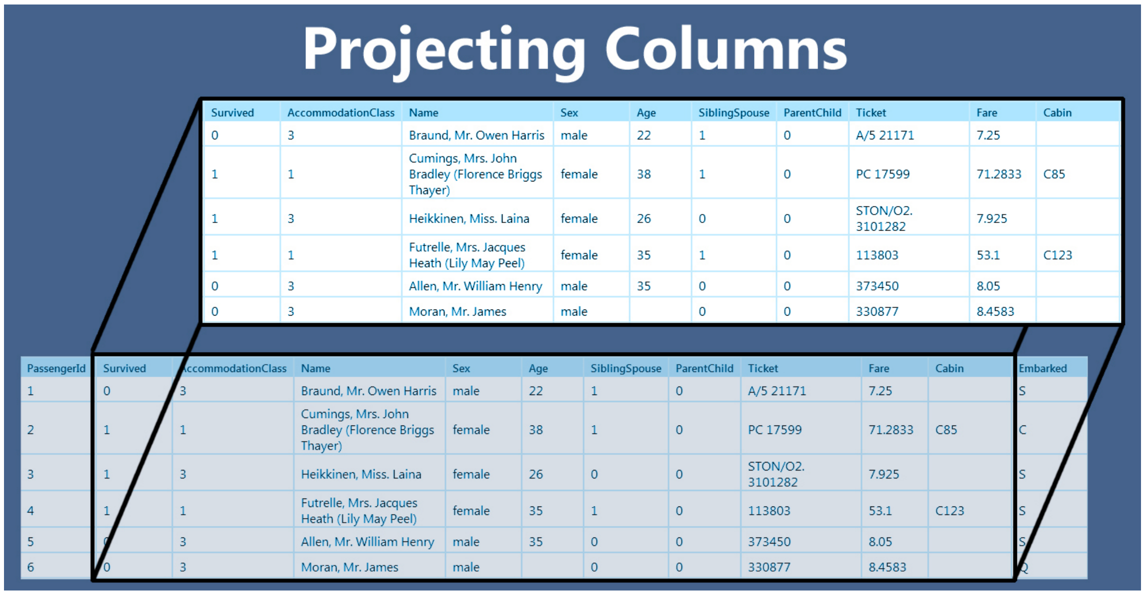
compare to Age

Figure 4.26: Survived plots

4.3.6 Exercise: Renaming Your Columns

Some of the column names in the dataset are abbreviated. To make the data more readable, we can rename columns like “Pclass”, “SibSp”, and “Parch”.

1. We will continue building off of the other modules already in your workspace. In the left menu, search for **Project Columns**, drag the module into your workspace and connect it to the **Metadata Editor**.



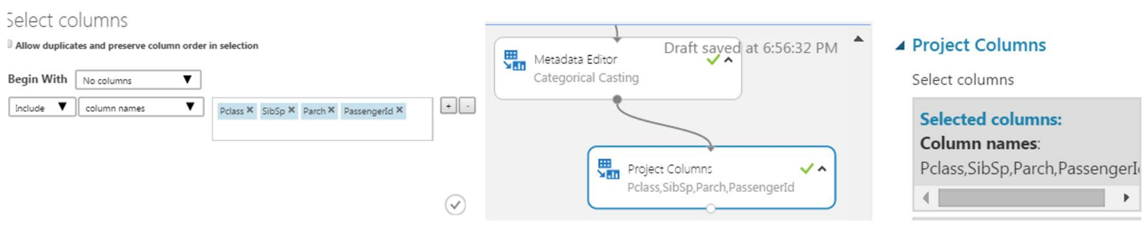
Projecting Columns

Survived	AccommodationClass	Name	Sex	Age	SiblingSpouse	ParentChild	Ticket	Fare	Cabin
0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	
1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85
1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925	
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123
0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05	
0	3	Moran, Mr. James	male		0	0	330877	8.4583	

PassengerId	Survived	AccommodationClass	Name	Sex	Age	SiblingSpouse	ParentChild	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q

Figure 4.27: In this example of projecting columns, all columns except for “Embarked” and “PassengerID” are being projected, essentially dropping these columns.

2. Select “Launch column selector” and choose “PassengerId”, “Pclass”, “SibSp”, and “Parch” in the textbox, leaving all of the other options unchanged (Figure: 4.28).



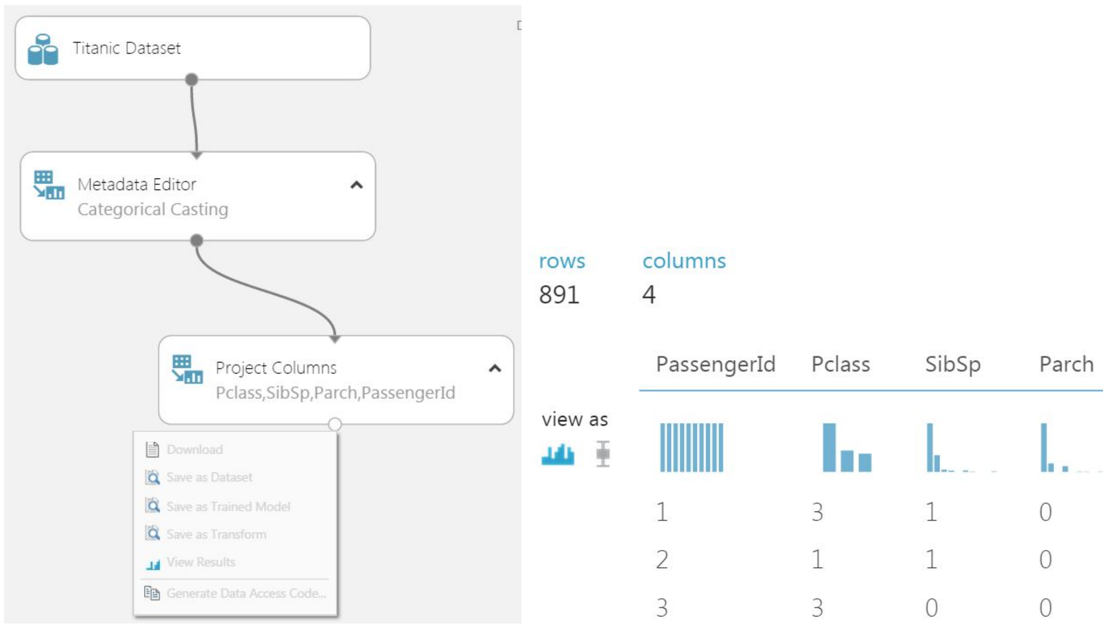
Select columns
Allow duplicates and preserve column order in selection
Begin With: No columns
Include | column names | Pclass ✕ SibSp ✕ Parch ✕ PassengerId ✕

Draft saved at 6:56:32 PM
Metadata Editor
Categorical Casting
Project Columns
Pclass,SibSp,Parch,PassengerId

Project Columns
Select columns
Selected columns:
Column names:
Pclass,SibSp,Parch,PassengerI

Figure 4.28: Project Columns, column selector

3. Label your **Project Columns** module “PassengerId, Pclass, Sibsp, Parch”, **Run** the workspace, right-click on the bottom middle node of the **Project Columns** module and select “Visualize” (Figure: 4.29)

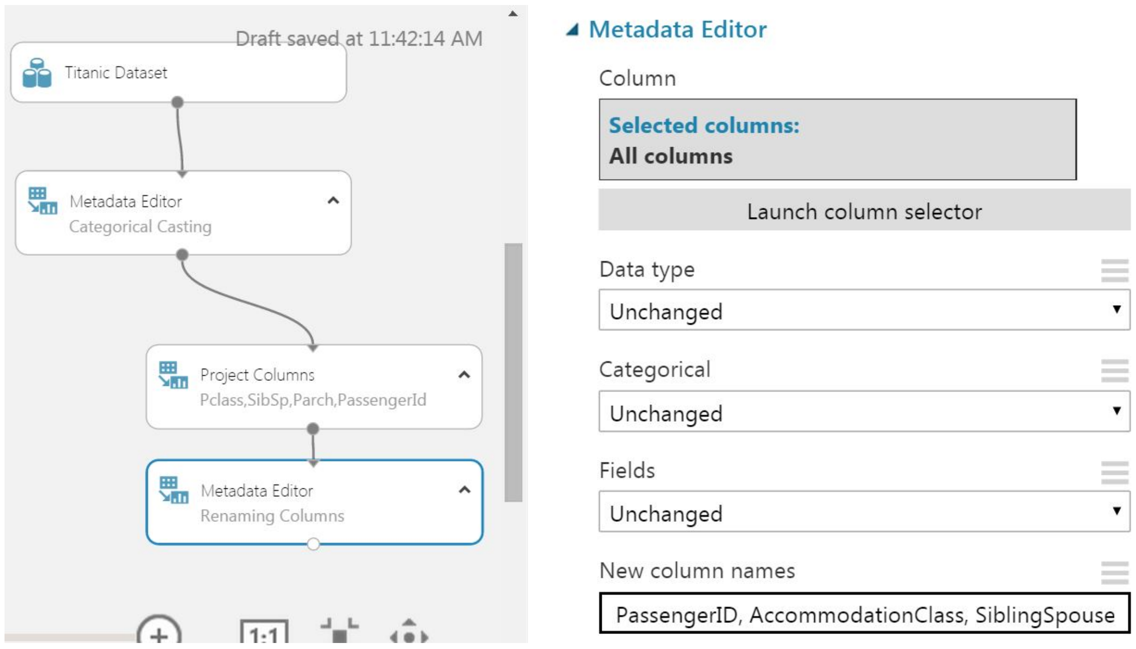


rows 891 columns 4

PassengerId	Pclass	SibSp	Parch
1	3	1	0
2	1	1	0
3	3	0	0

Figure 4.29: Project Columns visualization

4. In the left menu, search for another **Metadata Editor** and drag the module into your workspace. Connect the **Metadata Editor** to the **Project Columns** module.
5. Select "Launch column selector" and change "Begin With" to "All Columns". Directly below "Begin With" you will see a "+" and "-" sign. Remove the existing parameter by selecting the "-" sign, then submit your options.
6. In the **Metadata Editor** parameters under "New column names", enter the column names in the order that corresponds to the data, separated by commas (e.g. PassengerId, AccommodationClass, SiblingSpouse, ParentChild). Leave all other options in the editor unchanged (Figure: 4.30) and label it "Renaming Columns".



Metadata Editor

Column

Selected columns:
All columns

Data type: Unchanged

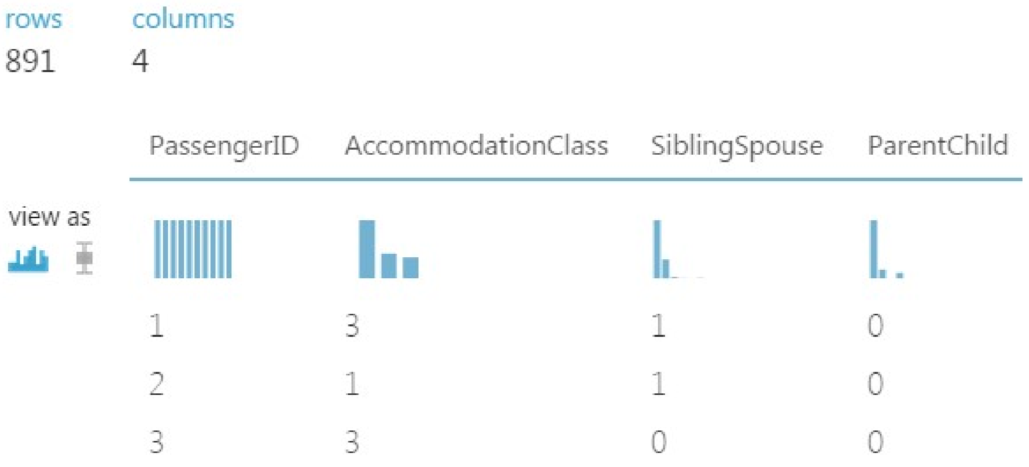
Categorical: Unchanged

Fields: Unchanged

New column names: PassengerID, AccommodationClass, SiblingSpouse

Figure 4.30: Metadata Editor settings

7. Right-click on the **Metadata Editor** and select "Visualize". Verify that the new column names are correct.



rows 891 columns 4

PassengerID	AccommodationClass	SiblingSpouse	ParentChild
1	3	1	0
2	1	1	0
3	3	0	0

Figure 4.31: Metadata Editor visualization

4.3.7 Exercise: Joining Tables

In the previous exercise, we specified certain columns from the larger dataset to edit. Now we must rejoin our changes with the existing dataset.

1. In the left menu, search for another **Project Columns** module and connect it to the **Metadata Editor** labeled "Categorical Casting""". There will now be two **Project Columns** modules parallel to one another, both connecting to the same **Metadata Editor** (Figure: 4.32).

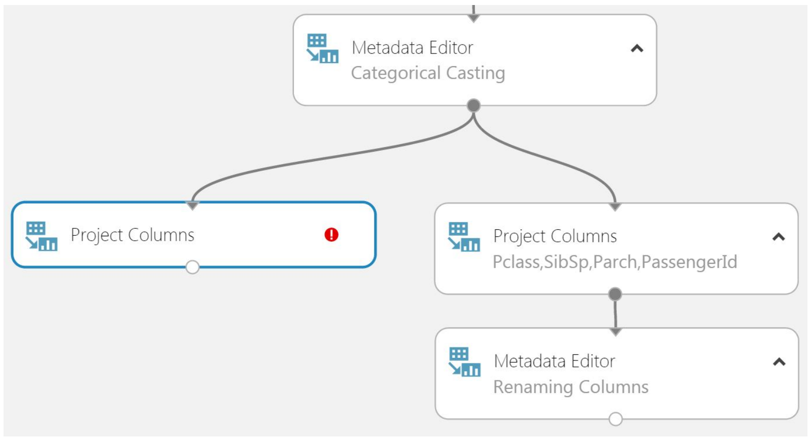
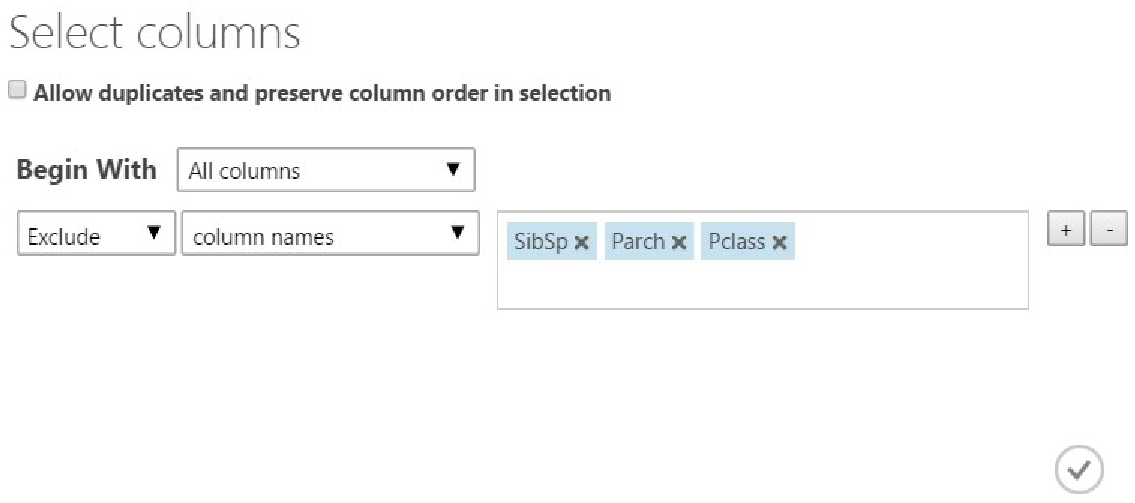


Figure 4.32: Connect the Metadata Editor

2. In the **Project Columns** module select "Launch column selector". Make changes so that "Begin With" is set to "All Columns", "include" is changed to "exclude", and in the textbox choose "SibSp", "Parch", and "Pclass" for removal. Do not remove "PassengerId" as we will need this for the join. Label the module "Exclude SibSp, Parch, Pclass" (Figure: 4.33).


Select columns

Allow duplicates and preserve column order in selection

Begin With: All columns

Exclude | column names | SibSp, Parch, Pclass

Figure 4.33: Launch column selector

3. In the left menu, search for the **Join** module and drag it into your workspace.
4. Connect the top left node of the **Join** module to the **Project Columns** module labeled "Exclude SibSp, Parch, Pclass". Connect the top right node of the **Join** module to the **Metadata Editor** labeled "Renaming Columns" (Figure: 4.34).

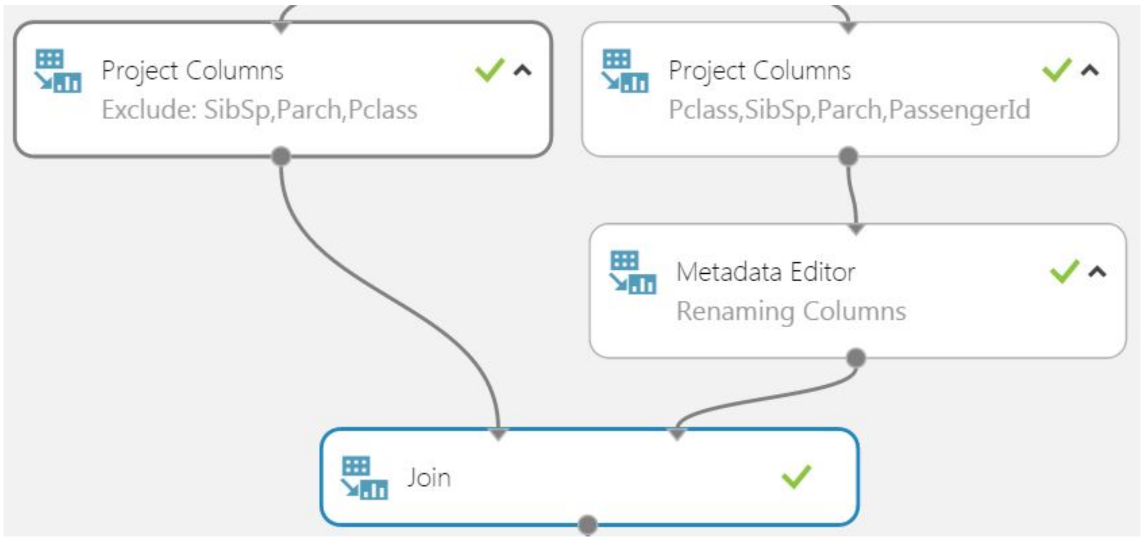
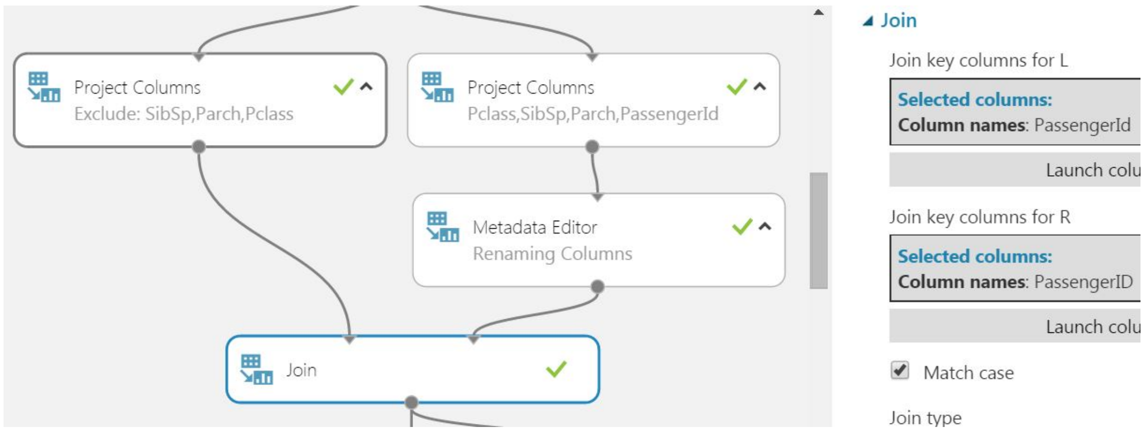


Figure 4.34: Connect the Join module

5. In the **Join** module options under "Join key columns for L", select "Launch column selector", input "PassengerId" in the textbox, and submit your selections. Under "Join key columns for R" repeat the same process, selecting "PassengerId" again.
6. In order to prevent "PassengerId" from appearing twice as a result of the join, uncheck the box for "Keep right key columns in" (Figure: 4.35)



Join

Join key columns for L

Selected columns:
Column names: PassengerId

Launch colu

Join key columns for R

Selected columns:
Column names: PassengerID

Launch colu

✓ Match case

Join type

Figure 4.35: Join module settings

7. **Run** the experiment, right-click the **Join** module and select "Visualize". Verify that the tables have joined properly by checking the "PassengerId" column.

4.3.8 Exercise: Using Descriptive Statistics and Cleaning Missing Data

Cleaning missing values from large datasets is necessary in creating a model. Before we can know what to scrub, we must identify the missing values. We will use the **Descriptive Statistics** feature to provide us with this information.

1. From the left menu, search for the **Descriptive Statistics** module and connect it to the bottom

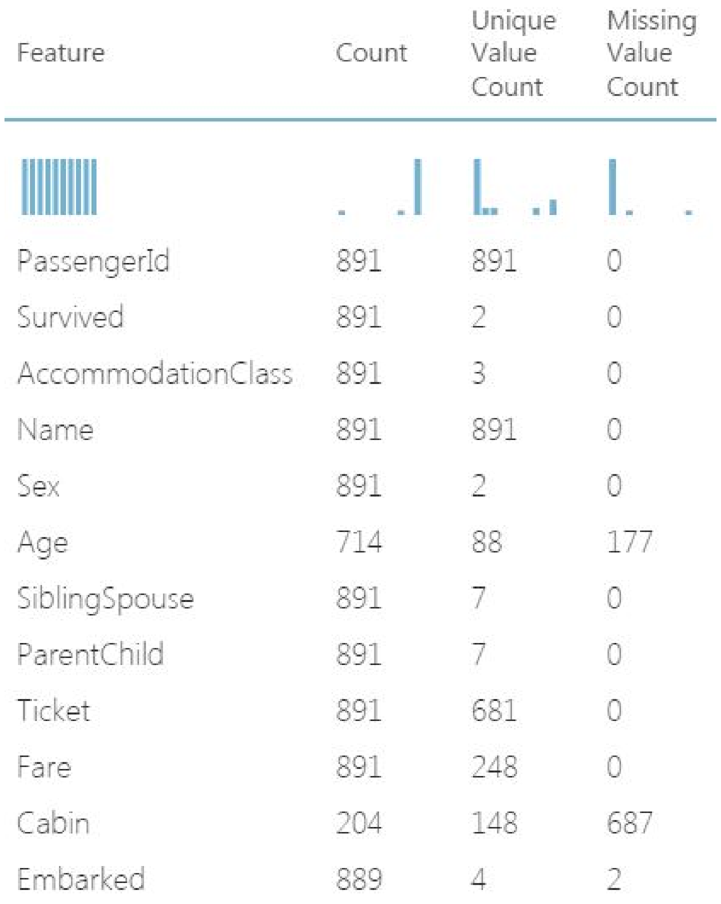
of the existing **Join** module.

2. **Run** the experiment, right-click on the **Descriptive Statistics** module and select "Visualize"""" to view the output (Figure: 4.36).

What do you notice about the information provided? What was the mean age of travellers on the Titanic? What was the age of the oldest person? The youngest? How do you interpret the youngest age? What was the lowest fare price? The median?

Look at the categories of data. Identify the columns that you think hold little data mining value for what you might want to use the dataset for. For example, "PassengerId" is only a primary key, holding no value other than being a unique identifier. The same applies for "Name" and "Ticket", both holding little value in this context.

Next, look at the "Missing Value Count" column. Which columns have missing values? For example "Age" has 177 missing values.

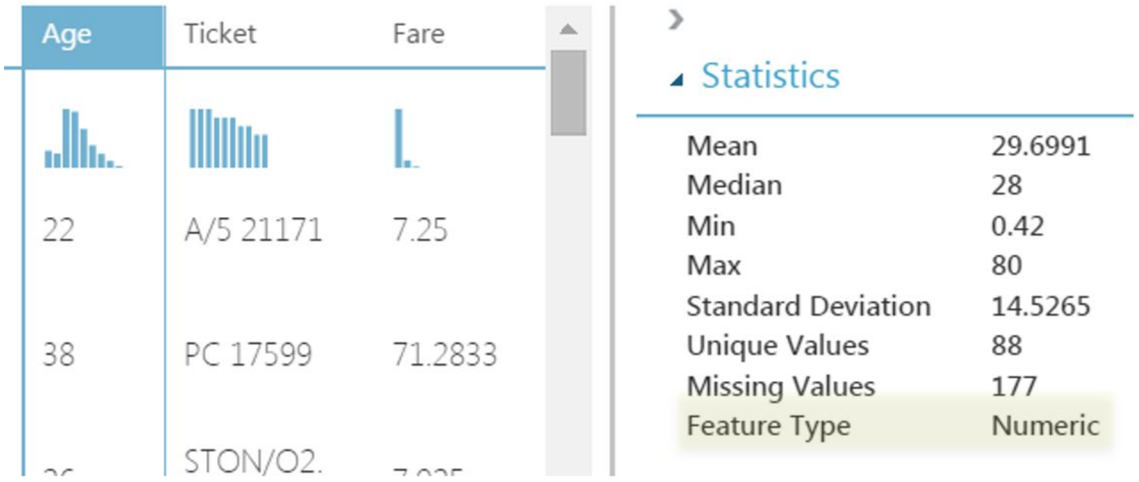


Feature	Count	Unique Value Count	Missing Value Count
PassengerId	891	891	0
Survived	891	2	0
AccommodationClass	891	3	0
Name	891	891	0
Sex	891	2	0
Age	714	88	177
SiblingSpouse	891	7	0
ParentChild	891	7	0
Ticket	891	681	0
Fare	891	248	0
Cabin	204	148	687
Embarked	889	4	2

Figure 4.36: Descriptive Statistics visualization

3. Exit the **Descriptive Statistics** visualization and visualize the output of the **Join** module.

4. “Age”, “Cabin”, and “Embarked” all have missing values. For each of these columns we want to learn the feature type. To do this, select the corresponding column name in the visualization output. Expand the “Statistics” dropdown on the left and take note of the “Feature Type” (e.g. “Age” is Numeric). Repeat this step for each column (Figure: 4.37).



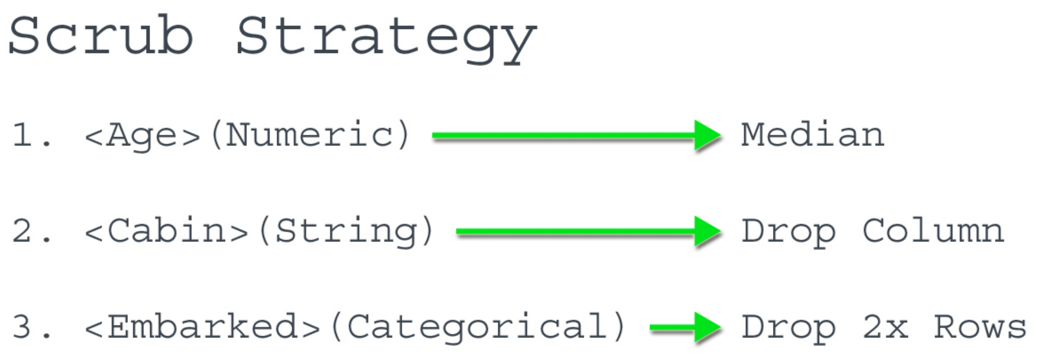
Mean	29.6991
Median	28
Min	0.42
Max	80
Standard Deviation	14.5265
Unique Values	88
Missing Values	177
Feature Type	Numeric

Figure 4.37: Feature type

Now that we have performed an analysis on the missing values and identified some of their features, we can develop a scrub strategy.

- “Age” has 177 missing values and is numeric. Dropping this column would result in a significant loss of data. However, since we know that all “Age” values are numeric, we can easily replace these missing values with the median of the dataset without too much information loss.
- “Cabin” has 687 missing values and is a string. Therefore it might be best to drop the column from our dataset.
- “Embarked” is only missing two values and is categorical. Because there aren’t many missing values, dropping the rows of missing values will not have a great impact on the data.

Now that we understand the impact of removing certain values, it is very important to consider the order of operations. Because “Cabin” has the most missing values, we should drop it first. Doing this will decrease the size of the table going forward, meaning less iteration and processing. After this we will replace the missing “Age” values with the median. In Azure ML, as in any data mining software, row drops should always happen last. Therefore the two missing rows from “Embarked” will be dropped last. We will call this attack plan, a scrub strategy (Figure: 4.38).

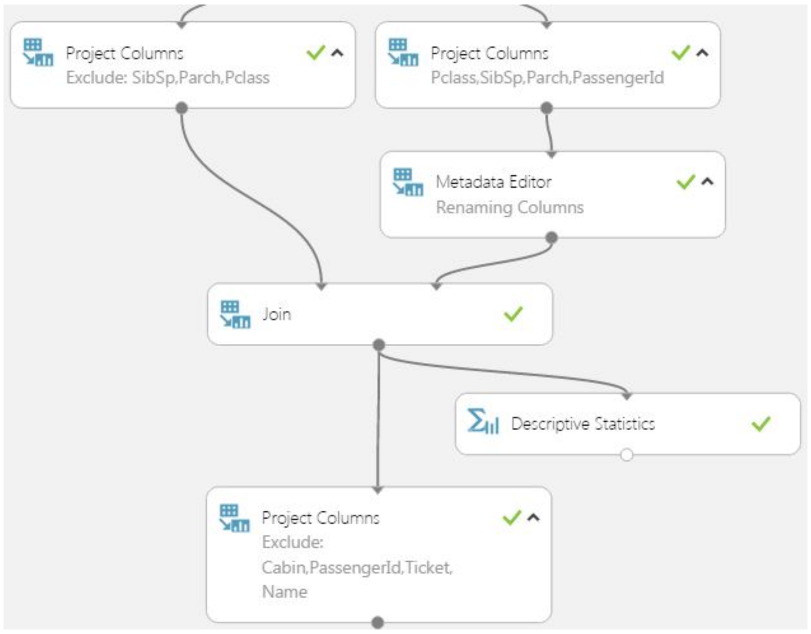


Scrub Strategy

1. <Age>(Numeric) → Median
2. <Cabin>(String) → Drop Column
3. <Embarked>(Categorical) → Drop 2x Rows

Figure 4.38: The approaches used on each attribute to clean missing values.

5. To drop the columns we previously identified as having little value ("Cabin", "PassengerId", "Ticket", "Name") we will use a **Project Columns** module. Search for this module from the left menu, drag it into the workspace, and connect it to the **Join** module (Figure: 4.39).



Project Columns
Exclude: SibSp,Parch,Pclass

Project Columns
Pclass,SibSp,Parch,PassengerId

Metadata Editor
Renaming Columns

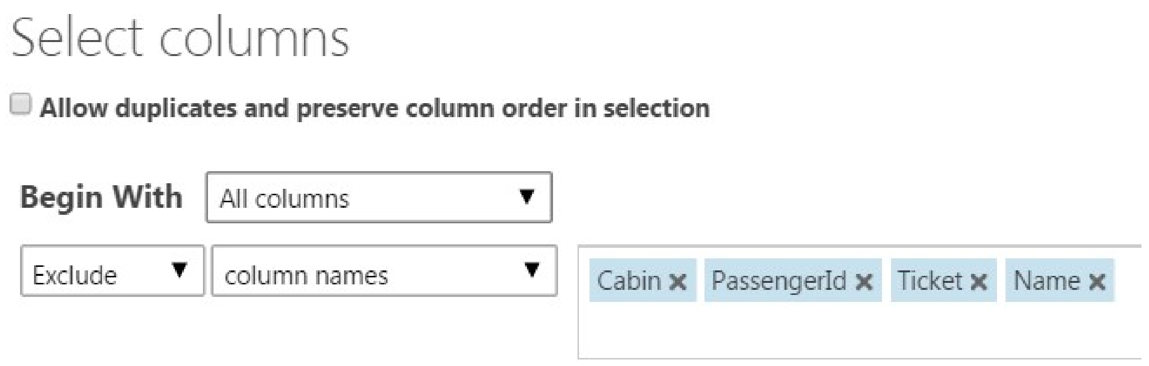
Join

Descriptive Statistics

Project Columns
Exclude:
Cabin,PassengerId,Ticket,
Name

Figure 4.39: Connect the Project Columns module

6. From the **Project Columns** module select "Launch column selector". Change "Begin With" to "All Columns", change "include" to "exclude", and select "Cabin", "PassengerId", "Ticket", and "Name" in the textbox (Figure: 4.40).



Select columns

Allow duplicates and preserve column order in selection

Begin With: All columns

Exclude | column names | Cabin ✕ PassengerId ✕ Ticket ✕ Name ✕

Figure 4.40: Column selector

7. To replace the missing "Age" values with the median, we will use a **Clean Missing Data** module. Search for the **Clean Missing Data** module in the left menu, drag it into the workspace, and connect it to the **Project Columns** module we just added (Figure: 4.41).

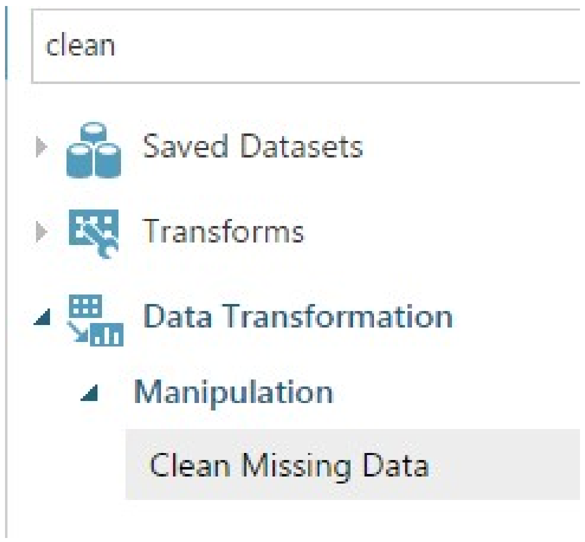
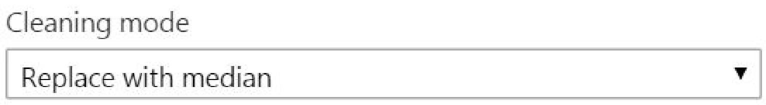


Figure 4.41: Clean Missing Data module

8. Select "Launch column selector" in the **Clean Missing Data** module. Set "For missing values" to "Replace with median" (Figure: 4.42).



Cleaning mode: Replace with median

Figure 4.42: Replace with median

Tip You can always check what median value will be replacing the missing values by returning to the **Descriptive Statistics** module.

9. To drop the remaining rows with missing values we will use the **Clean Missing Data** module again. Search for this module in the left menu, drag it into the workspace, and connect it to the previous **Clean Missing Data** module (Figure: 4.43)

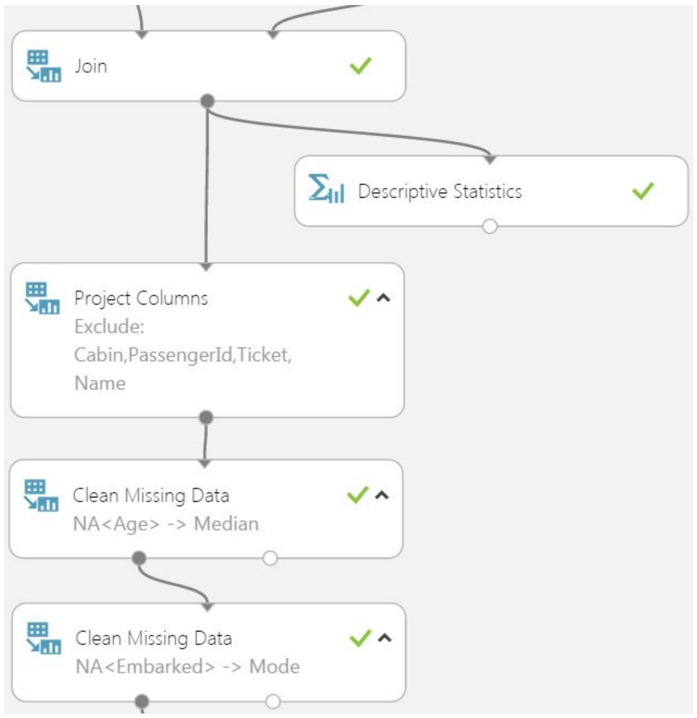


Figure 4.43: Connect the Clean Missing Data module

10. Select "Launch column selector" from within the module and set "For missing values" to "Remove entire row".
11. **Run** the experiment, right-click and visualize the output. Verify in the visualization that there are now only eight columns and 889 rows (Figure: 4.44)

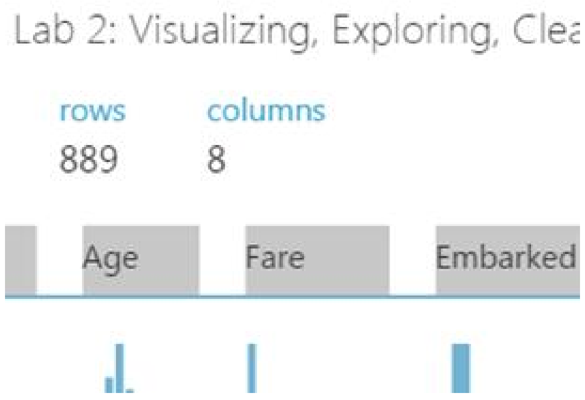


Figure 4.44: Visualization

After completing all of the above steps your workspace should look like the final result in Figure:

4.45. We are now ready to data mine!

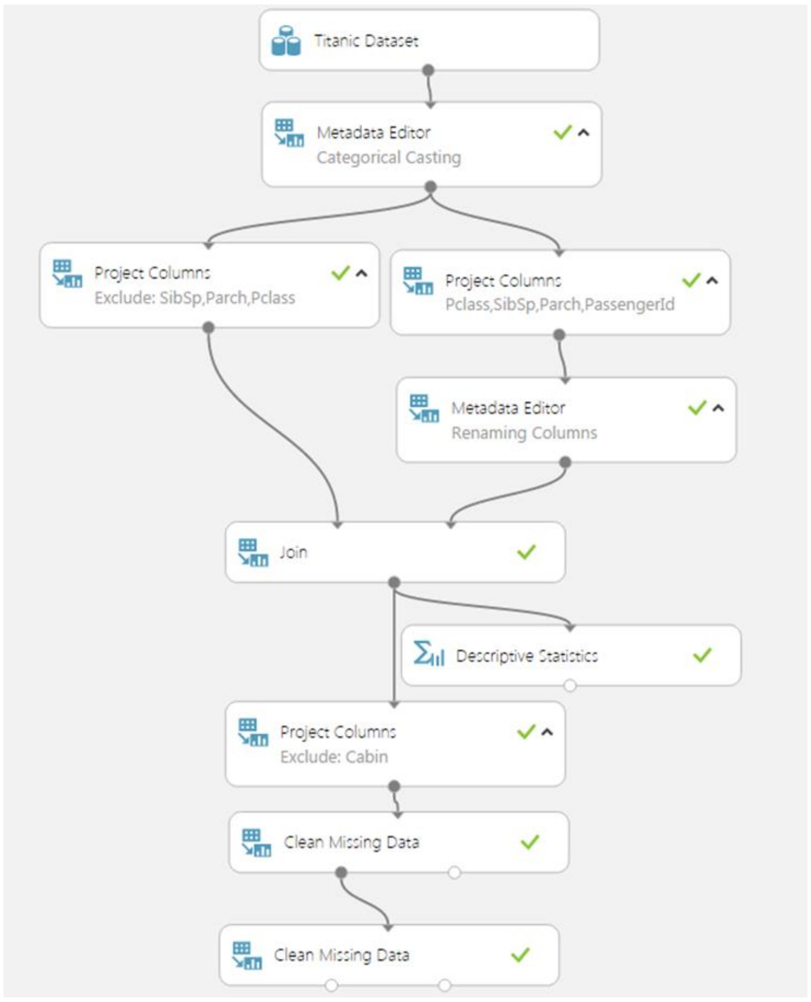


Figure 4.45: Final Result