Predict Drug for Patients according to their Health Metrics

[**Problem Statement**](#_3l8r47mg3fjc) **3**

[**Data Description**](#_nusffa56sgvx) **3**

[**Nodes In IBM SPSS Modeler Flow**](#_mavj80e45qty) **3**

[**Data Preparation**](#_mt5j0y9ghyxj) **5**

[Import Data Asset to Canvas](#_s76a37cyyq14) 5

[Run Data Audit](#_7j4uskytdwld) 5

[Run Type Node](#_z1ewgiamowjg) 6

[Adding a Table](#_n48j2oo2iprq) 7

[Creating a Distribution Graph](#_uvu0vspqcpax) 9

[Creating a Scatterplot](#_jbam3rt4xcuc) 12

[Deriving a New Field](#_fz66afvqcy55) 14

[**Building Models**](#_ogdnr54tumkg) **16**

[Split Train and Test Data](#_mrm8xkvmlxec) 17

[Building Model Using Decision Tree](#_mmta92mlbceh) 18

[Building Model Using K-Nearest Neighbor](#_sqi2rghgipd2) 23

[Using an Analysis Node](#_n13n0r34nzml) 28

[Building Model Using Neural Networks](#_z344oesb6sgf) 29

[Using an Analysis Node](#_csi7zrt5i8j5) 34

[**Modeler Flow**](#_lkrhttwez5f9) **35**

[**Implementing the Model with Best Model Accuracy**](#_iv16fd15ii8z) **36**

[**Deployment**](#_5110pqw8k38l) **38**

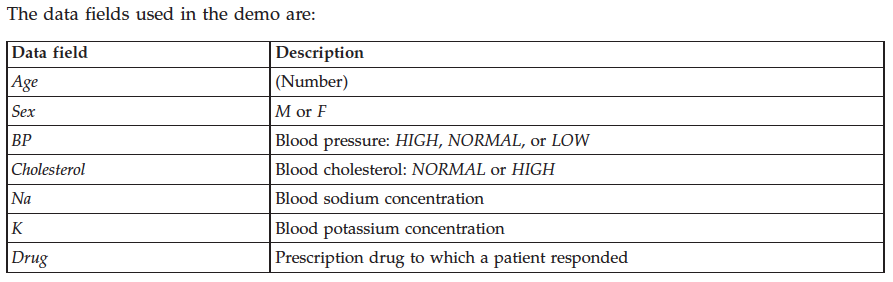
[Test](#_xarquk77hh2j) 40

[Implementation](#_f1l352a96l4v) 40

# Problem Statement

Imagine that you are a medical researcher compiling data for a study. You have collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of five medications. Part of your job is to use data mining to find out which drug might be appropriate for a future patient with the same illness. Predict drug for patients according to their health metrics.

# Data Description



# Nodes in IBM SPSS Modeler Flow

Before starting with the analysis, let’s have a look at different node options available in SPSS Modeler Flow.

On left side panel (Nodes Palette)you can see different types of nodes available for you to use while working on your data. There are six types of node categories:

1. **Record Operations:** As the name suggests, you can use them to perform operations such as selecting, appending, sorting on the record (row) level.
2. **Field Operations:** These nodes are helpful in the data preparation phase. You can filter data, rename features, and choose the type of your attributes.
3. **Graphs:** Nodes in this section will help you with basic data exploration and understanding distribution or relationship between features.
4. **Modeling:** These nodes provide different modeling algorithms for different types of problems.
5. **Outputs:** These nodes are helpful inunderstanding your dataand model. You can display results in table format or get a report on evaluation parameters of your model.
6. **Export:** After processing and modeling, this node will help you export datafrom the flow editor to your DSX project**.**

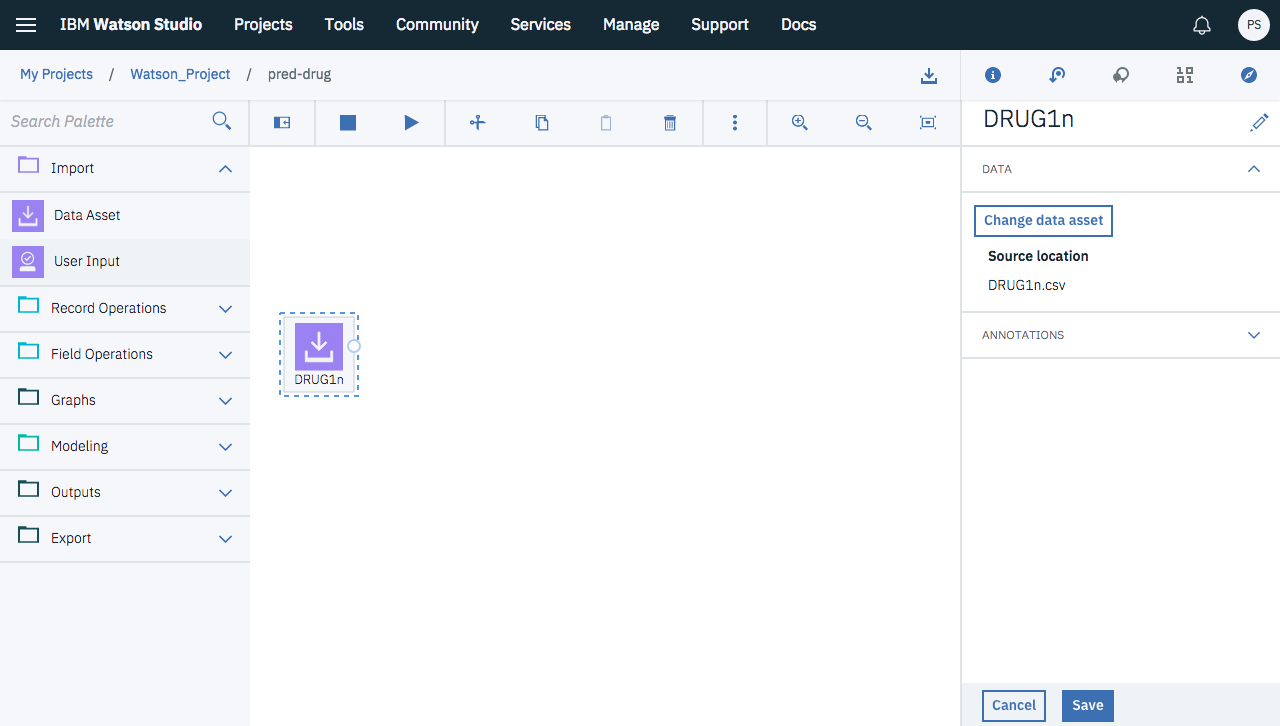
Drag and drop the node into the canvas and right-click to take further actions such as open, preview, or run.

# Data Preparation

To start working on the problem, first we need to get data into the canvas. It is as easy as drag-and-drop. To preview the file, right-click on a node and select **Preview**.

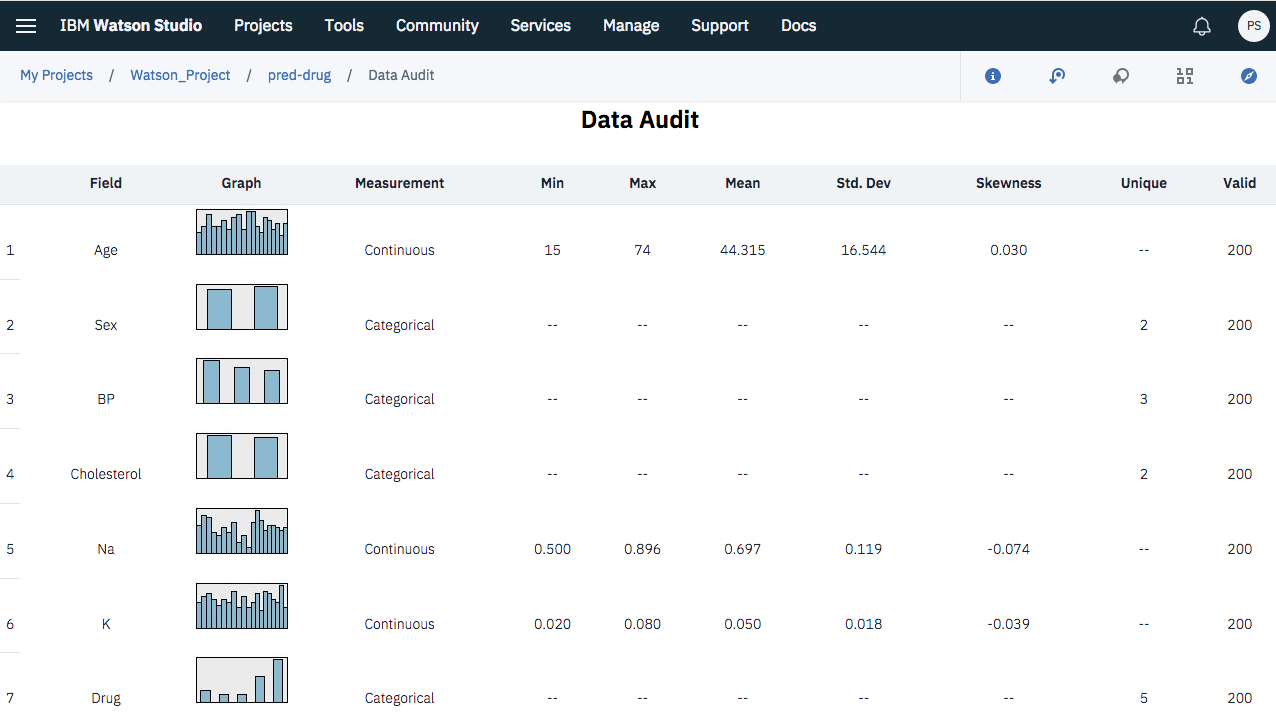
### **Import Data Asset to Canvas**

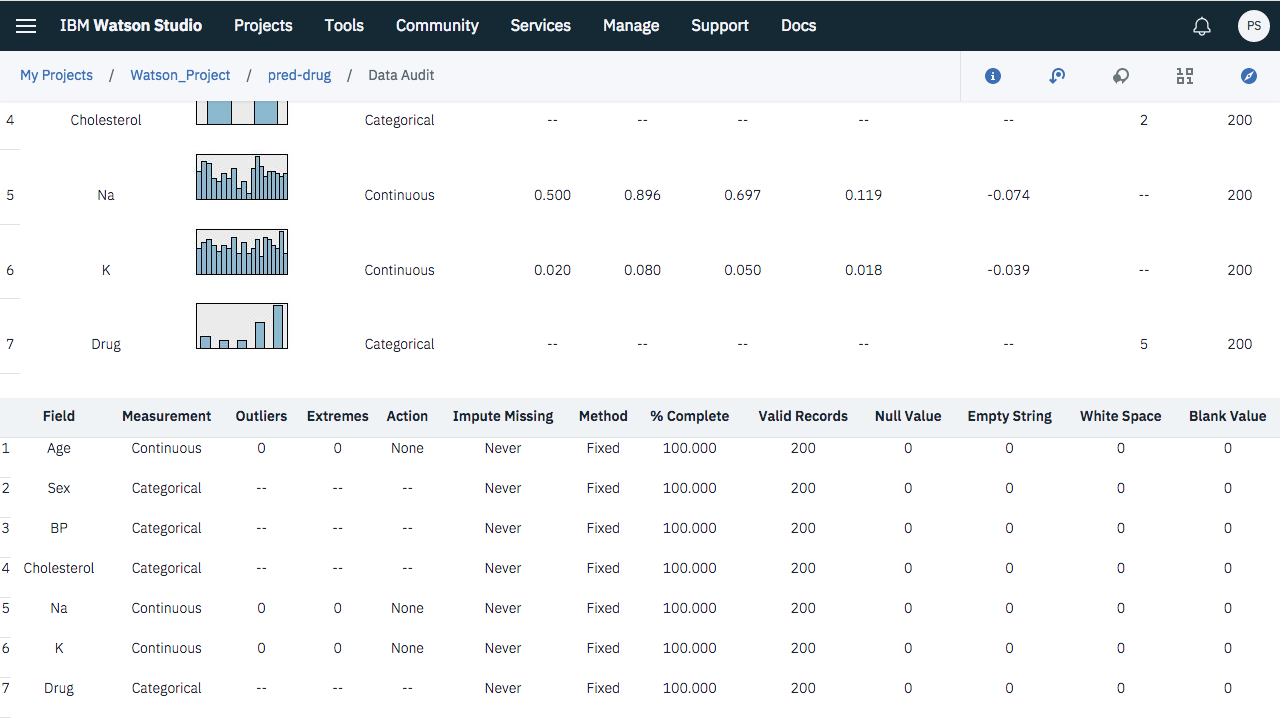
After you add the data set to your project you can find it in the **Data assets** section under the **Assets** tab. Once a Modeler Flow is created you can drag and drop Data Asset node and assign the data set.



### **Run Data Audit**

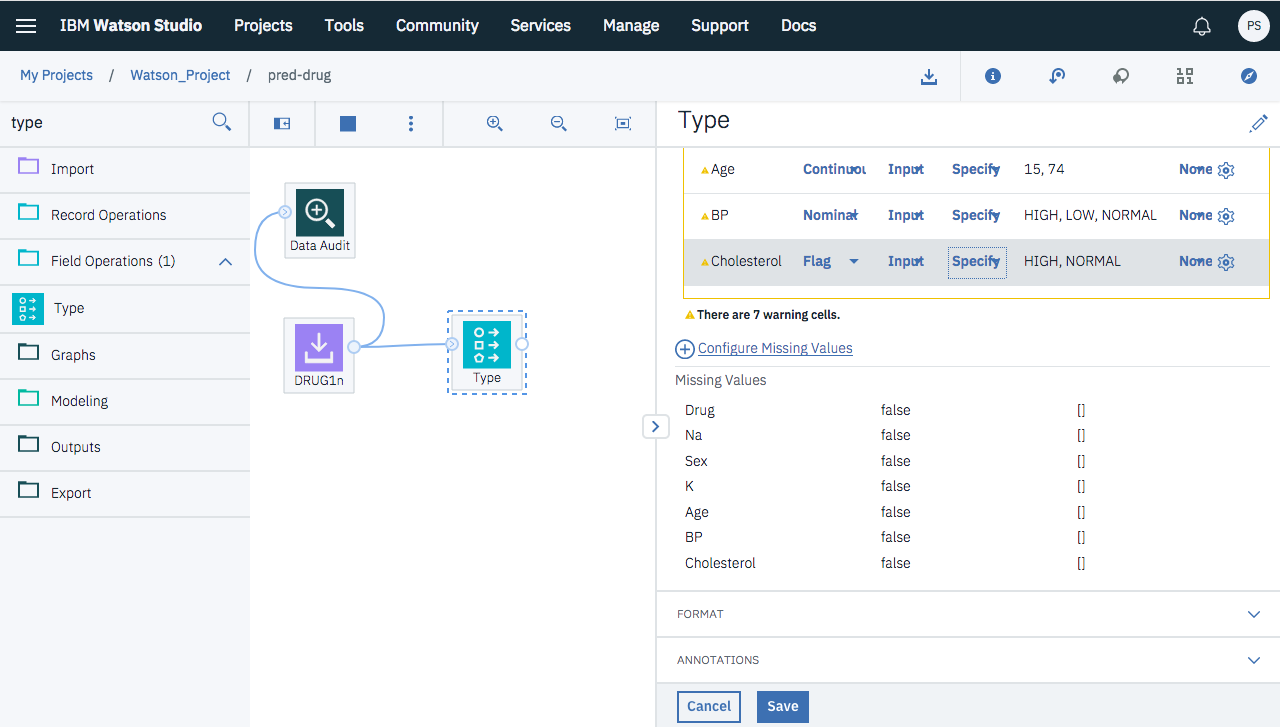
Add Data Audit node to go through the data set in detail. Once you add the node, connect it to the Data Asset node and then Run the Data Audit node. After running the node you can see your audit report on right side panel.





### **Run Type Node**

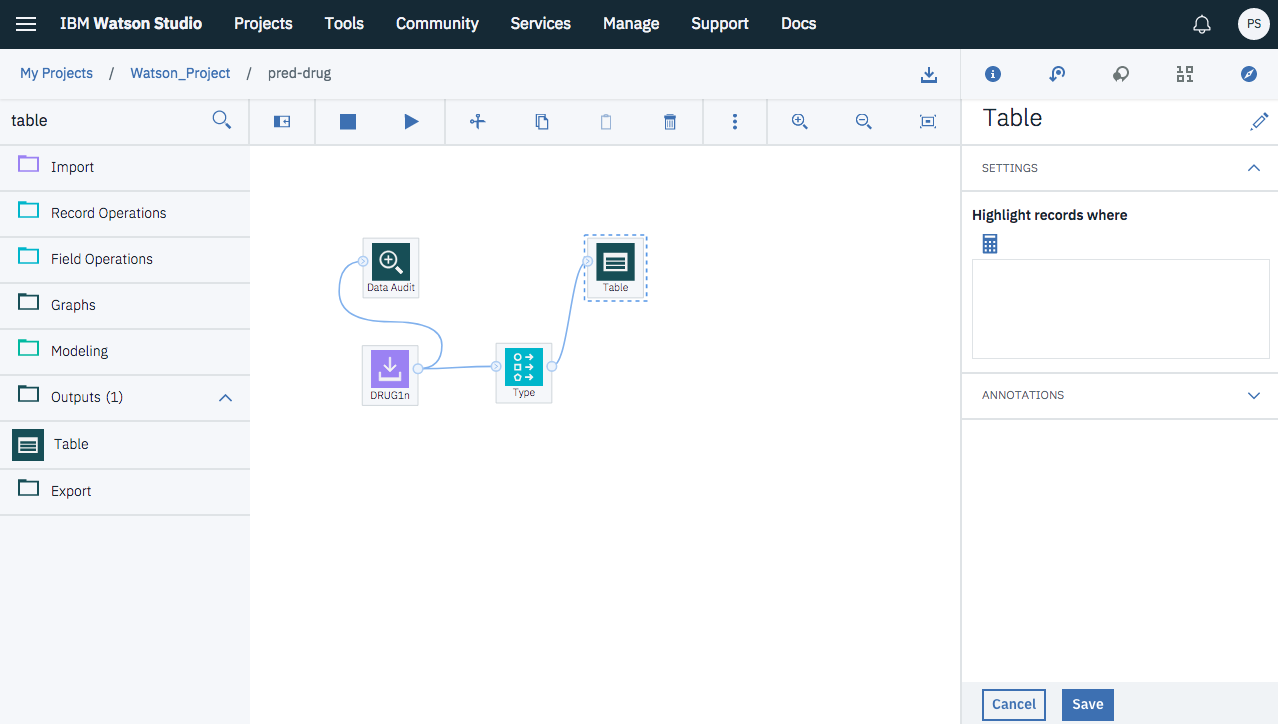
Drag the Type Node to check for missing values as well as assign types. After you connect type node to the Data Asset node, open the node. You can customize the node. Assign target variable and also configure missing values if any.

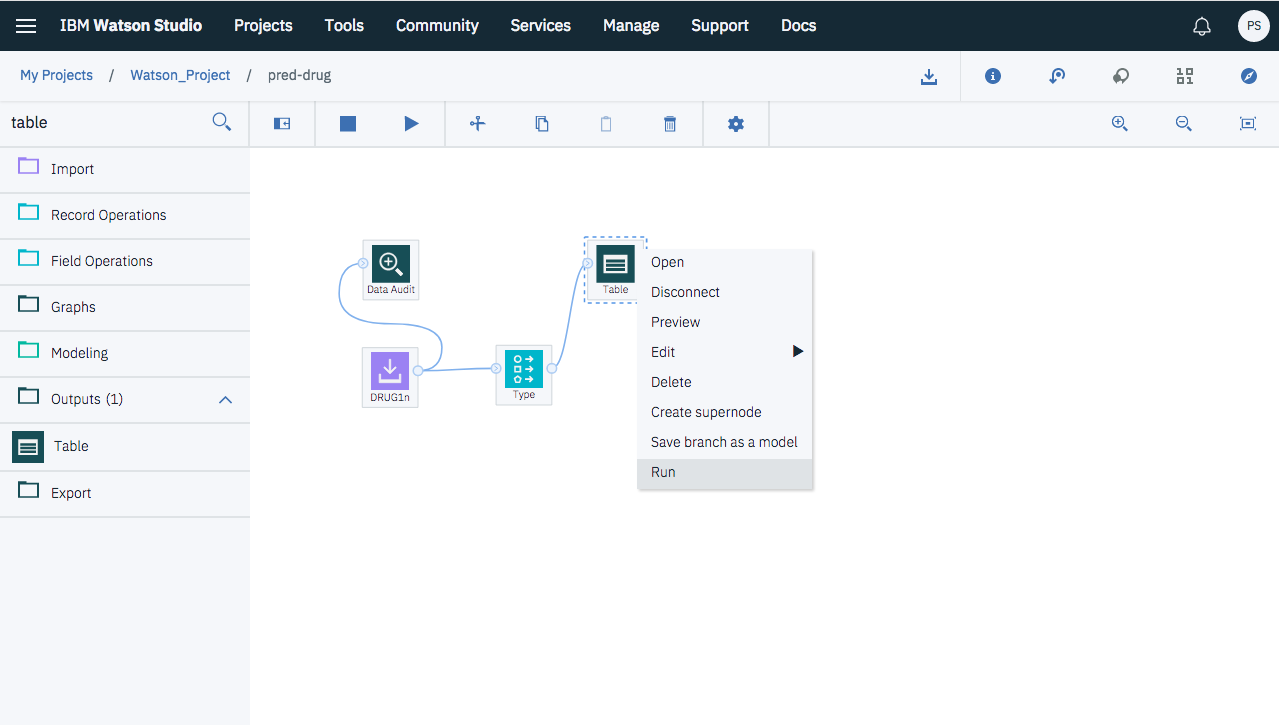


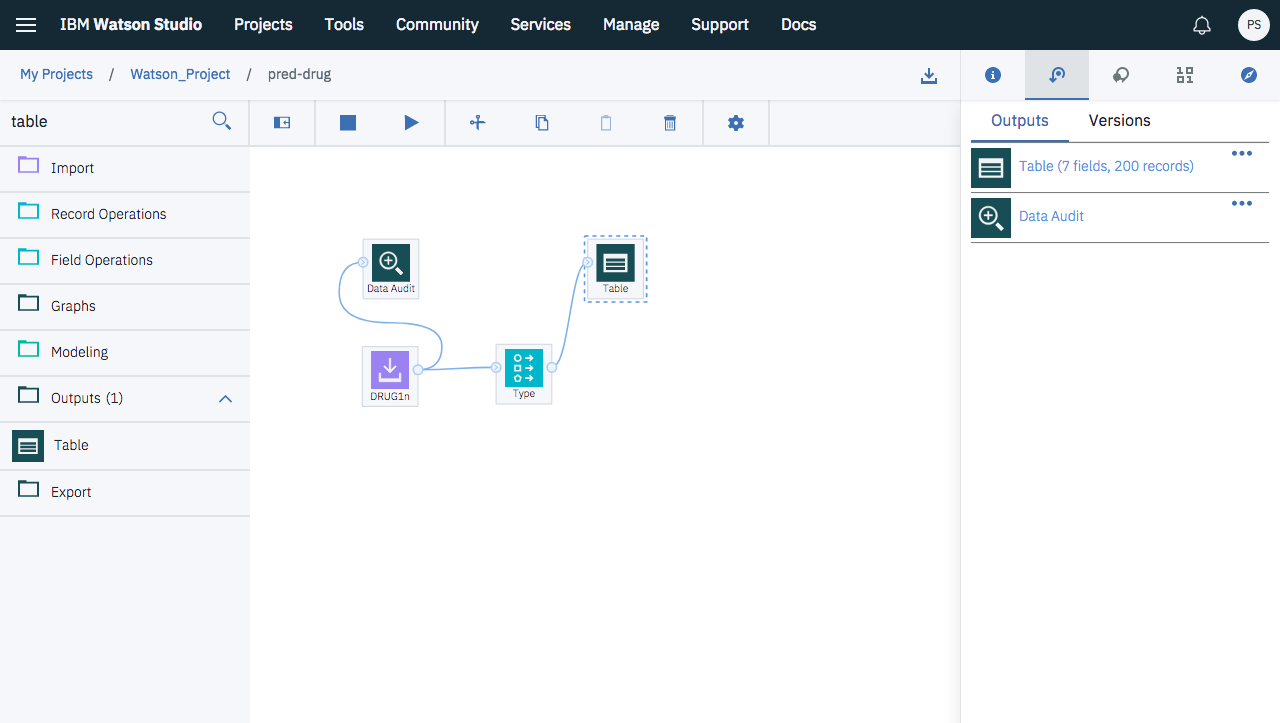
As we can see there are no missing values, we Save the Type Operations. You can also choose Read Values to view the actual values for each field based on the selections that you make from the Values column. This process is known as instantiation. After reading Clear Values and again Read to get the list of Missing Values.

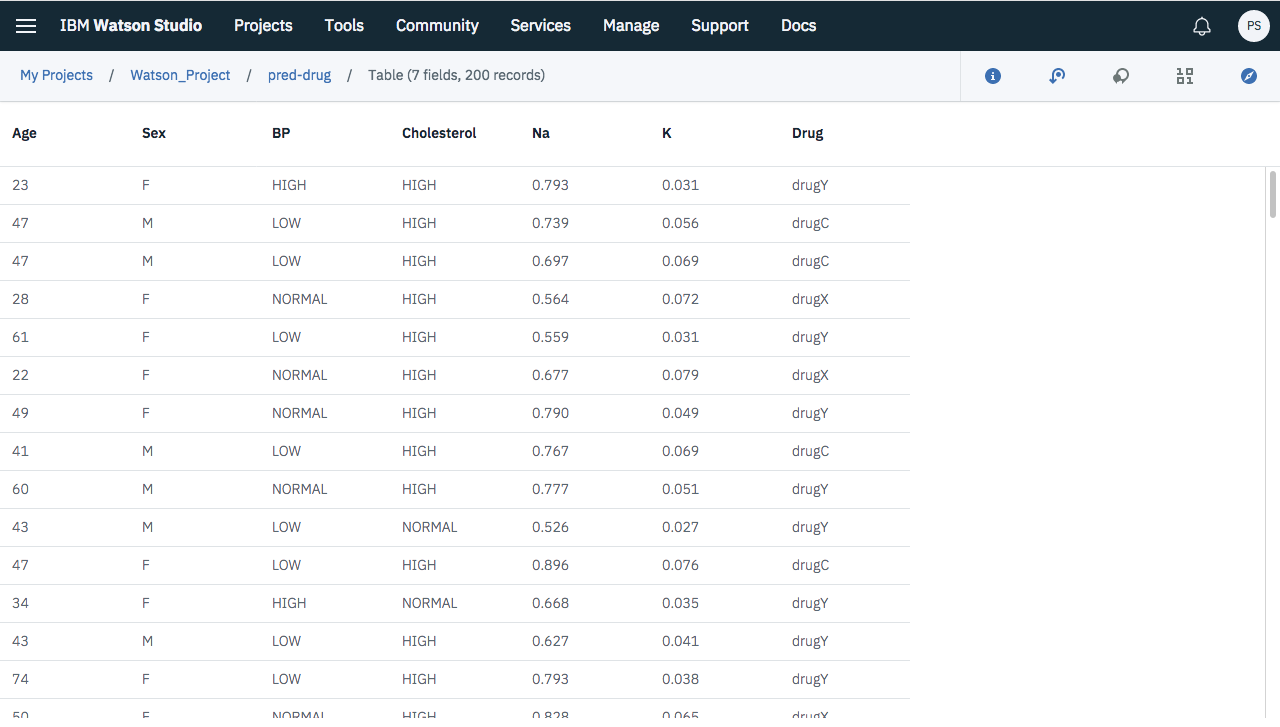
### **Adding a Table**

You can always add a table and check for the output of Type Node and later remove it if not required. After adding a Table node, connect it to Type node and Run it to view the output.



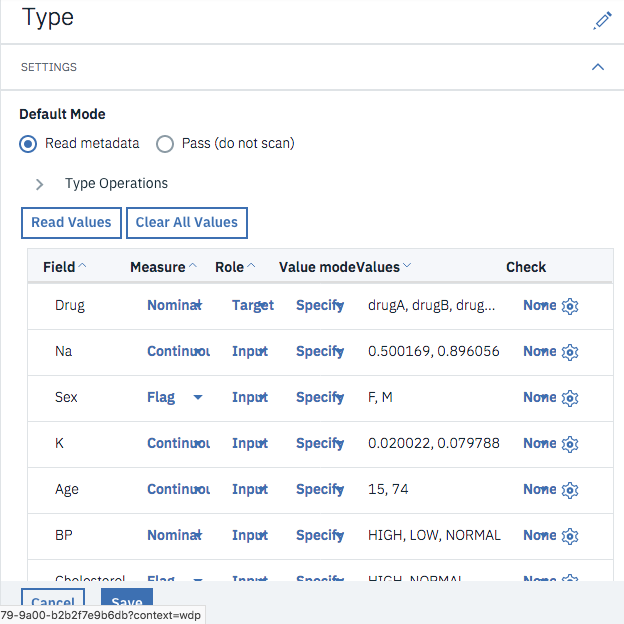


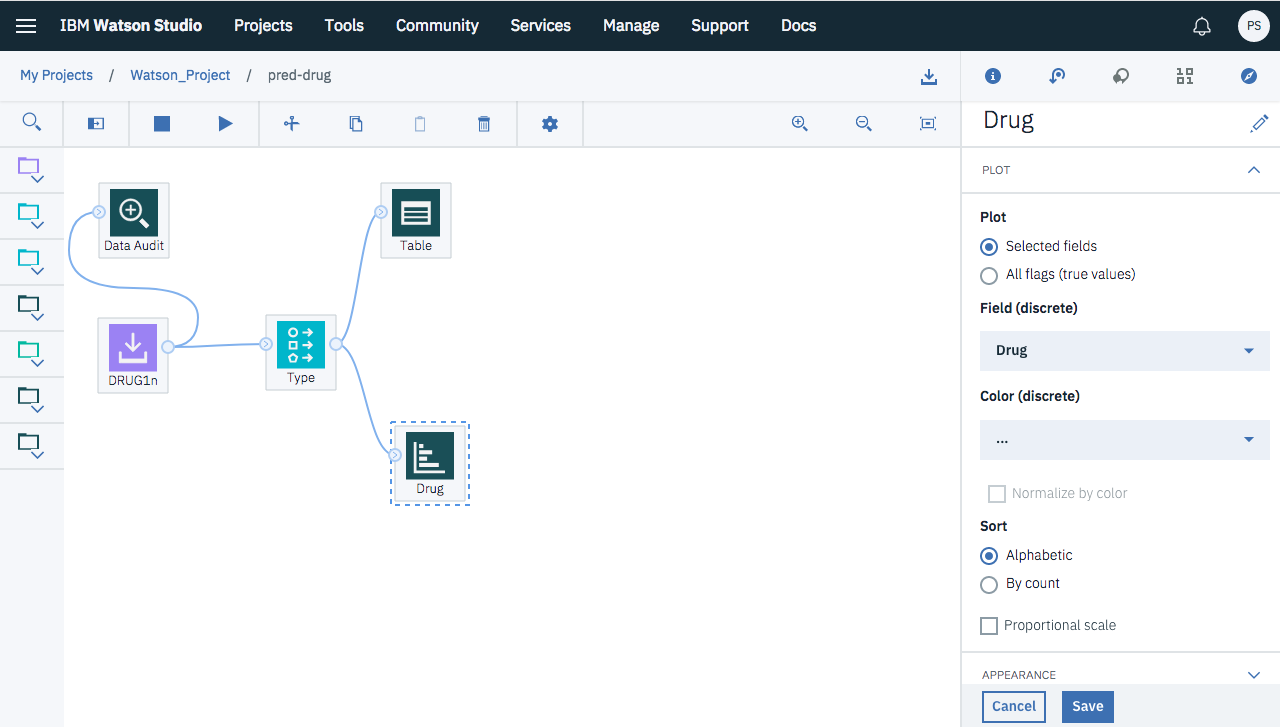


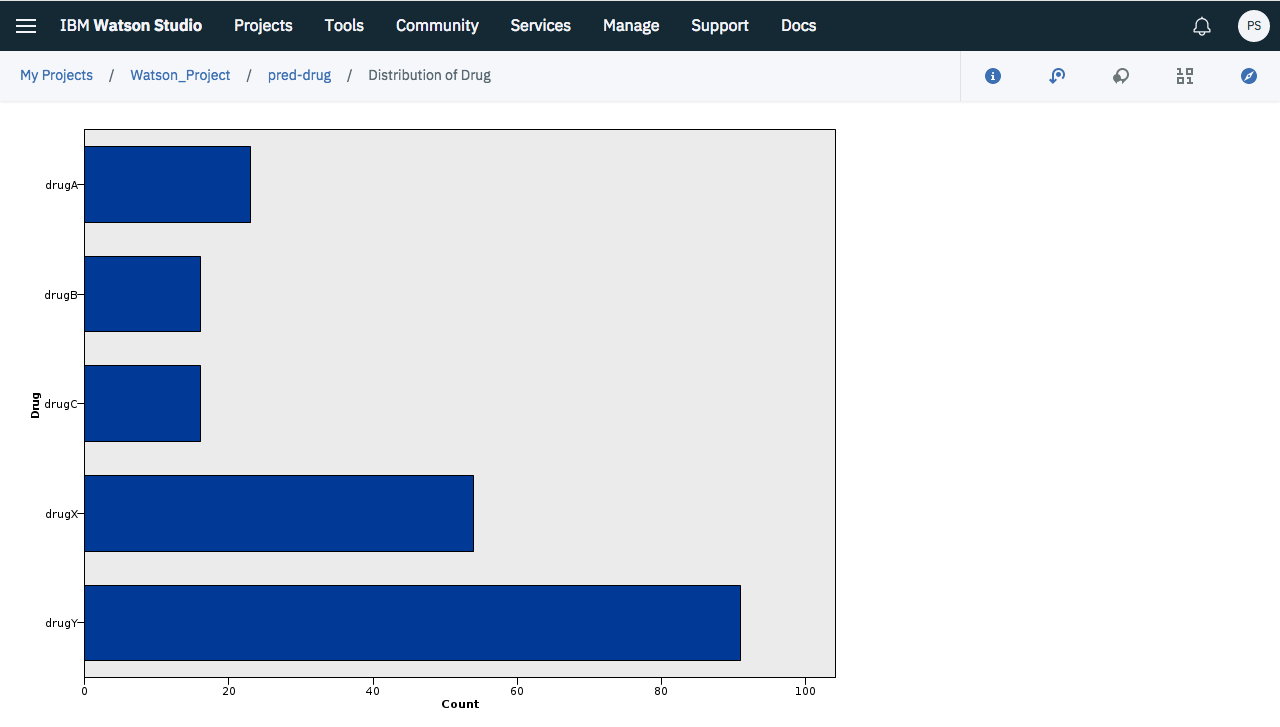


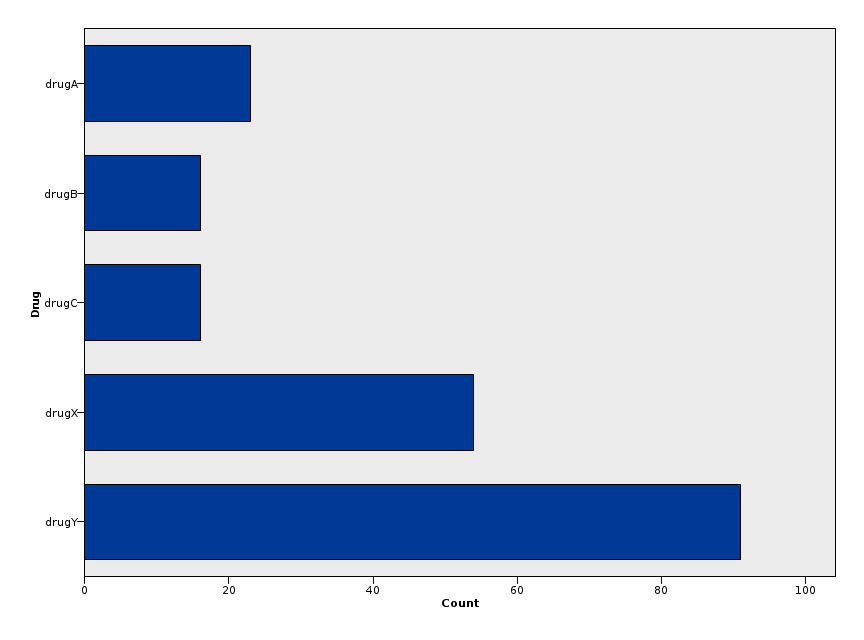
### **Creating a Distribution Graph**

During data mining, it is often useful to explore the data by creating visual summaries. IBM SPSS Modeler offers several different types of graphs to choose from, depending on the kind of data that you want to summarize. For example, to find out what proportion of the patients responded to each drug, use a Distribution node. Add a Distribution node to the stream and connect it to the Source node, then double-click the node to edit options for display. Select Drug as the target field whose distribution you want to show. Then, click Run from the dialog box.









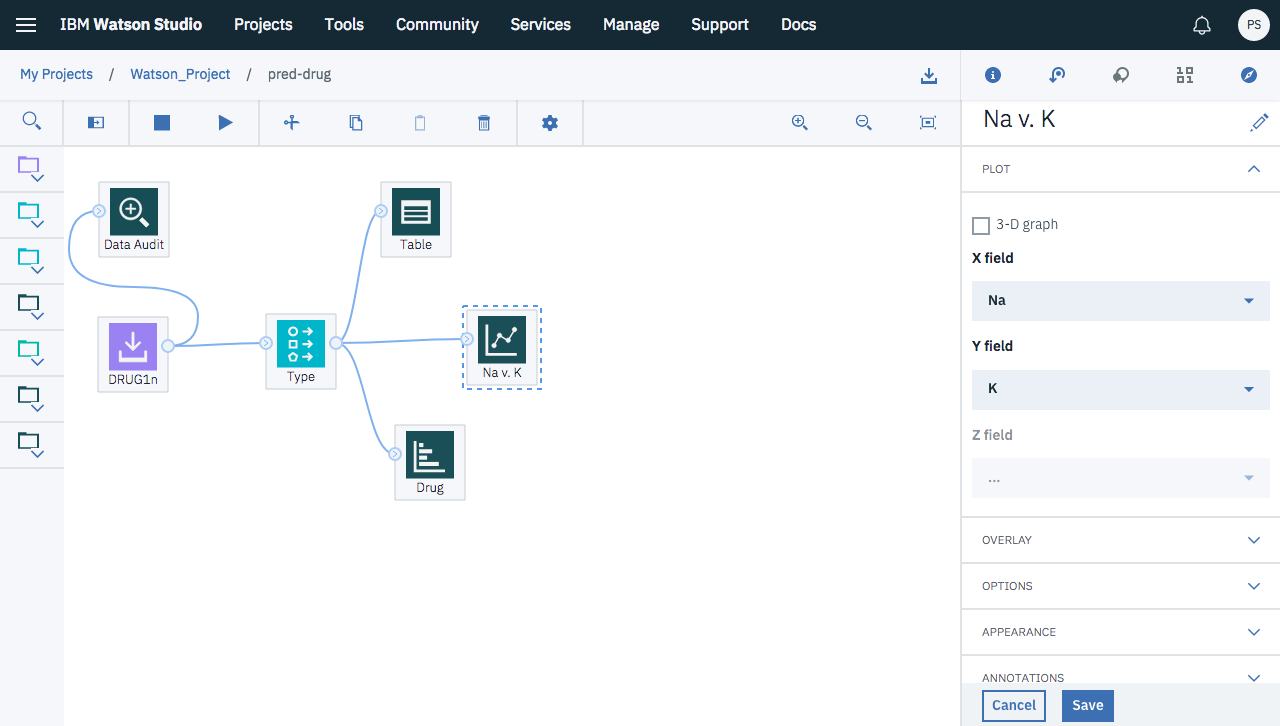
The resulting graph helps you see the "shape" of the data. It shows that patients responded to drug Y most often and to drugs B and C least often.

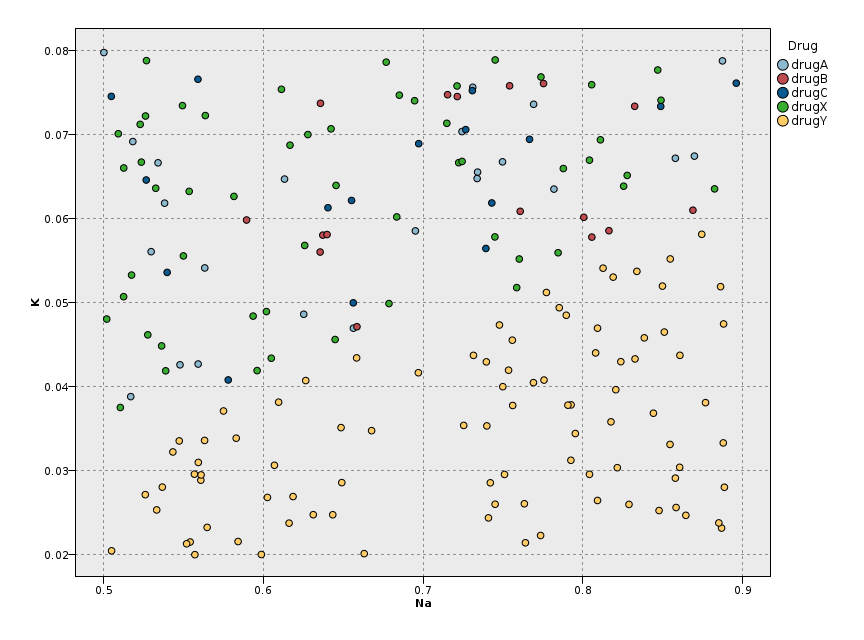
### **Creating a Scatterplot**

Now let's take a look at what factors might influence Drug, the target variable. As a researcher, you know that the concentrations of sodium and potassium in the blood are important factors. Since these are both numeric values, you can create a scatterplot of sodium versus potassium, using the drug categories as a color overlay. Place a Plot node in the workspace and connect it to the Source node, and double-click to edit the node.

On the Plot node, select Na as the X field, K as the Y field, and Drug as the overlay field. Then, click Run.

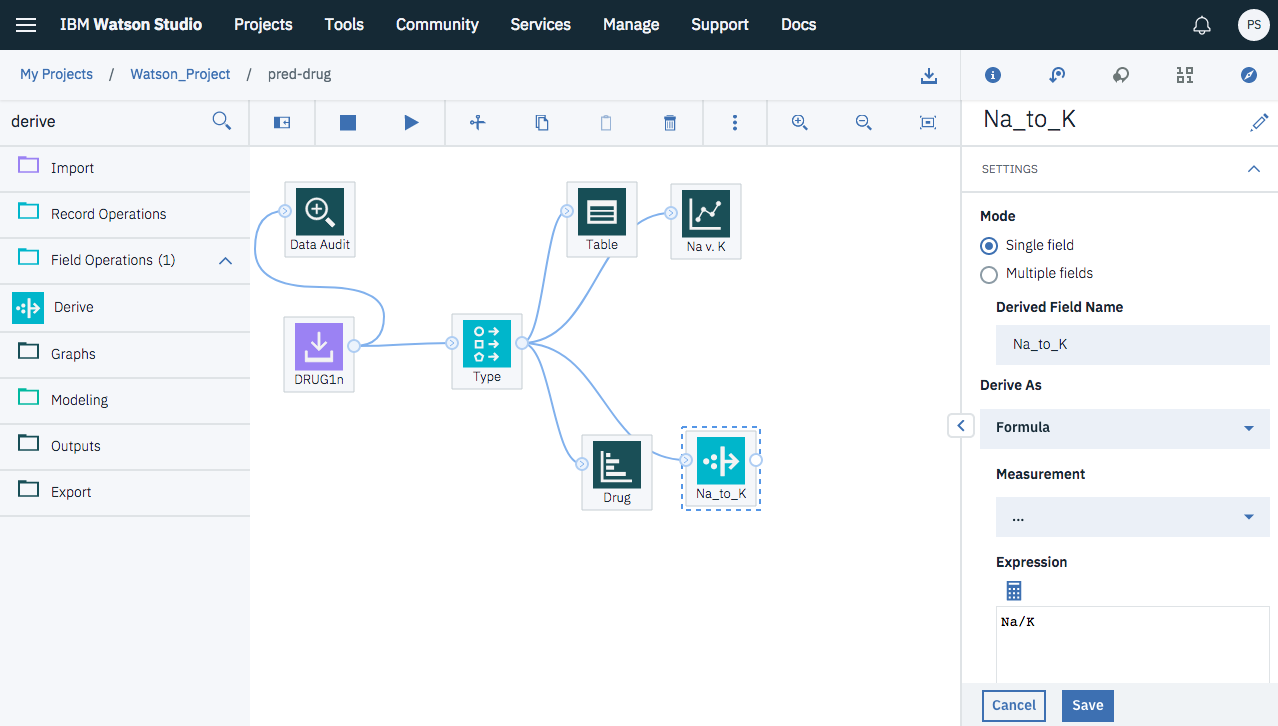
The plot clearly shows a threshold above which the correct drug is always drug Y and below which the correct drug is never drug Y. This threshold is a ratio--the ratio of sodium (Na) to potassium (K).



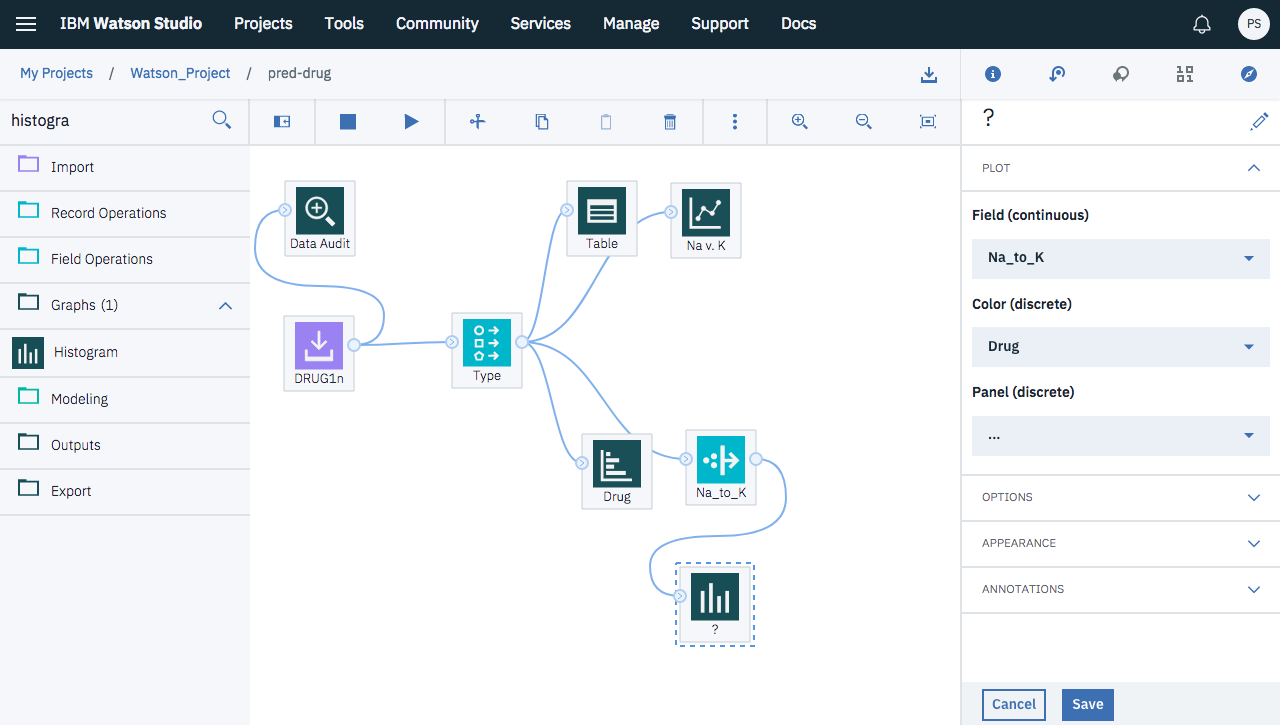


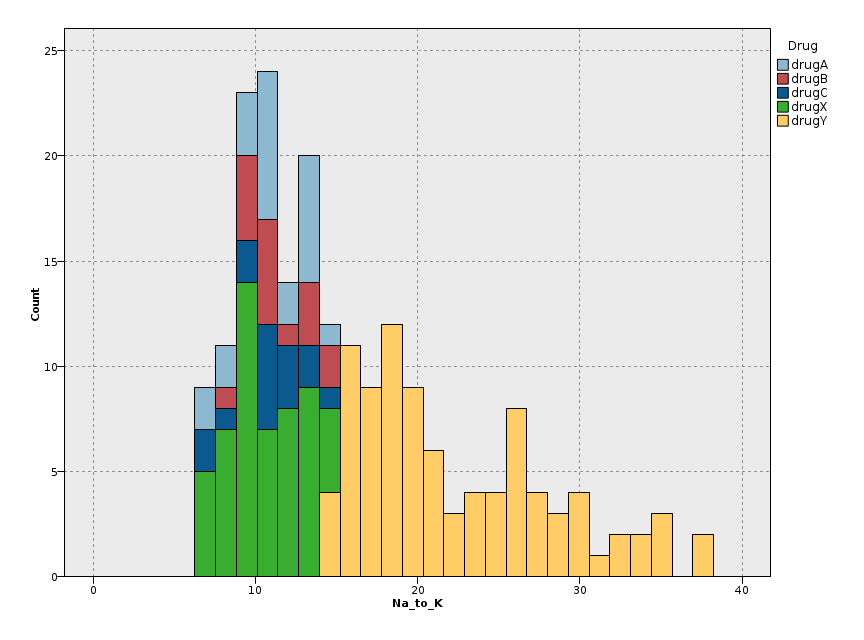
### **Deriving a New Field**

Since the ratio of sodium to potassium seems to predict when to use drug Y, you can derive a field that contains the value of this ratio for each record. This field might be useful later when you build a model to predict when to use each of the five drugs. Attach a Derive node to DRUG1n, then double-click the Derive node to edit it.



Name the new field Na\_to\_K. Since you obtain the new field by dividing the sodium value by the potassium value, enter Na/K for the formula. You can also create a formula by clicking the icon just to the right of the field. This opens the Expression Builder, a way to interactively create expressions using built-in lists of functions, operands, and fields and their values. You can check the distribution of your new field by attaching a Histogram node to the Derive node. In the Histogram node dialog box, specify Na\_to\_K as the field to be plotted and Drug as the overlay field.

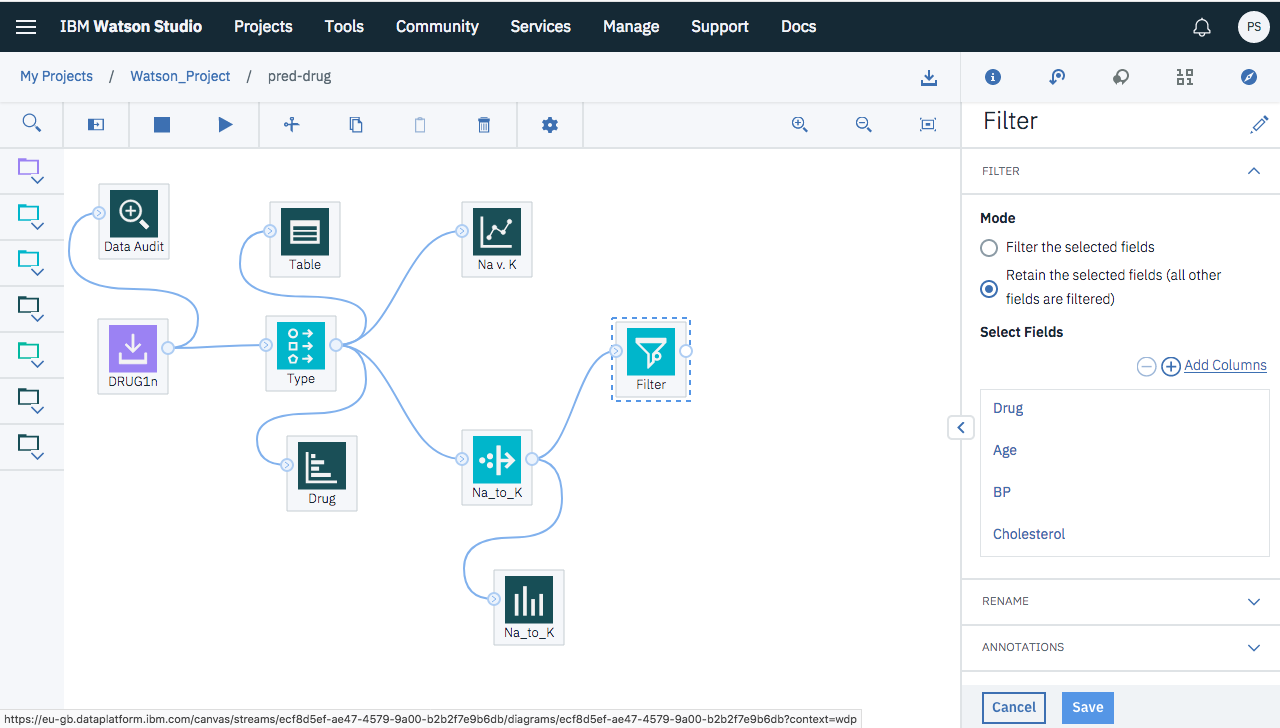


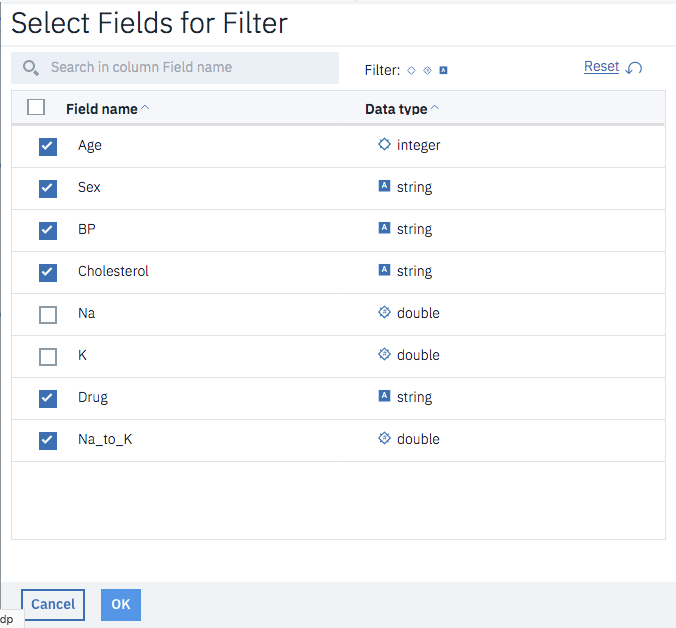


When you run the stream, you get the graph shown here. Based on the display, you can conclude that when the Na\_to\_K value is about 15 or above, drug Y is the drug of choice.

# Building Models

By exploring and manipulating the data, you have been able to form some hypotheses. The ratio of sodium to potassium in the blood seems to affect the choice of drug, as does blood pressure. But you cannot fully explain all of the relationships yet. This is where modeling will likely provide some answers. In this case, you will use try to fit the data using a rule-building model, C5.0. Since you are using a derived field, Na\_to\_K, you can filter out the original fields, Na and K, so that they are not used twice in the modeling algorithm. You can do this using a Filter node.

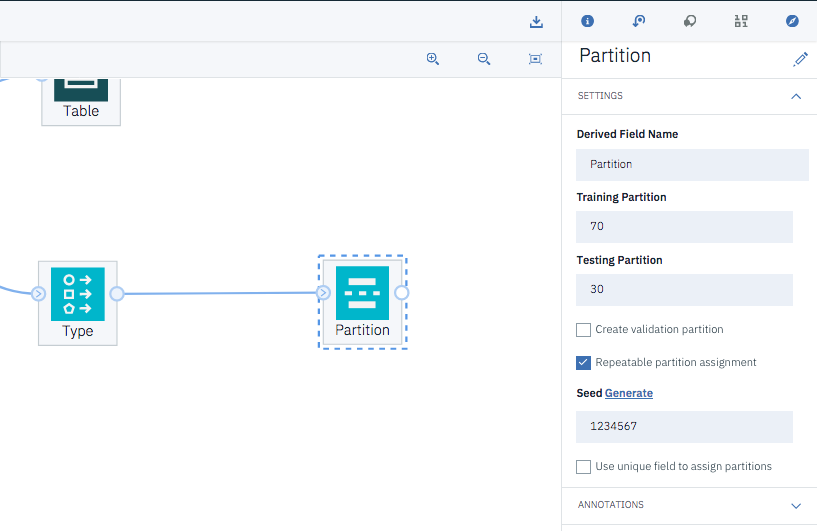




To estimate the model, place a C5.0 node in the workspace and attach it to the end of the stream as shown. Then click Run to run the stream.

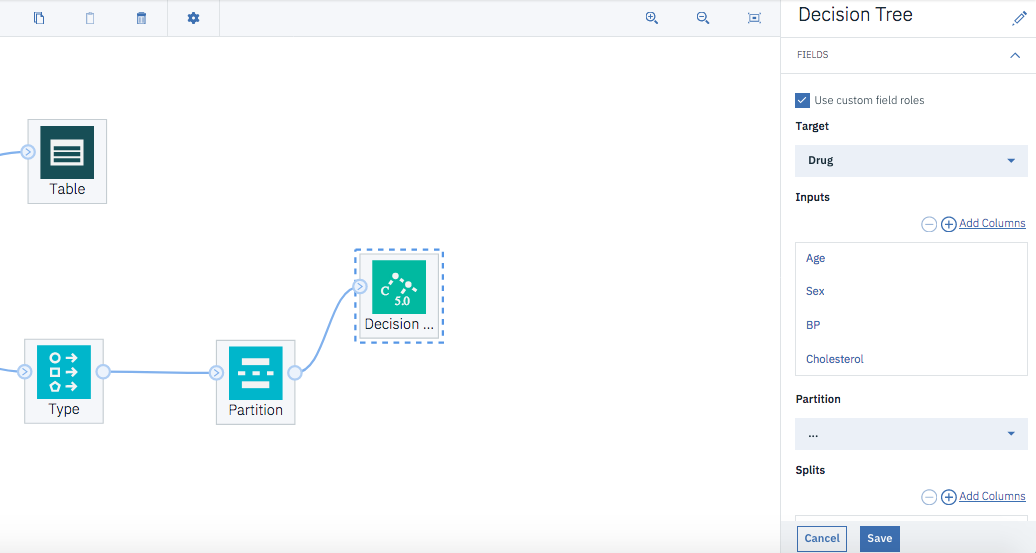
### **Split Train and Test Data**

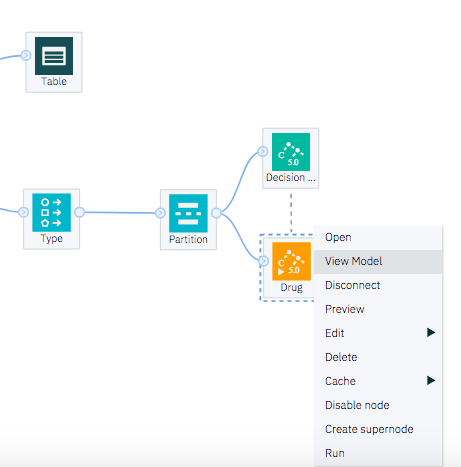
Let us split the data into train and test data using the Partition Node. Here 70% of the data will be considered as Training Partition and the remaining 30% as Test Partition. You can also add a validation set.

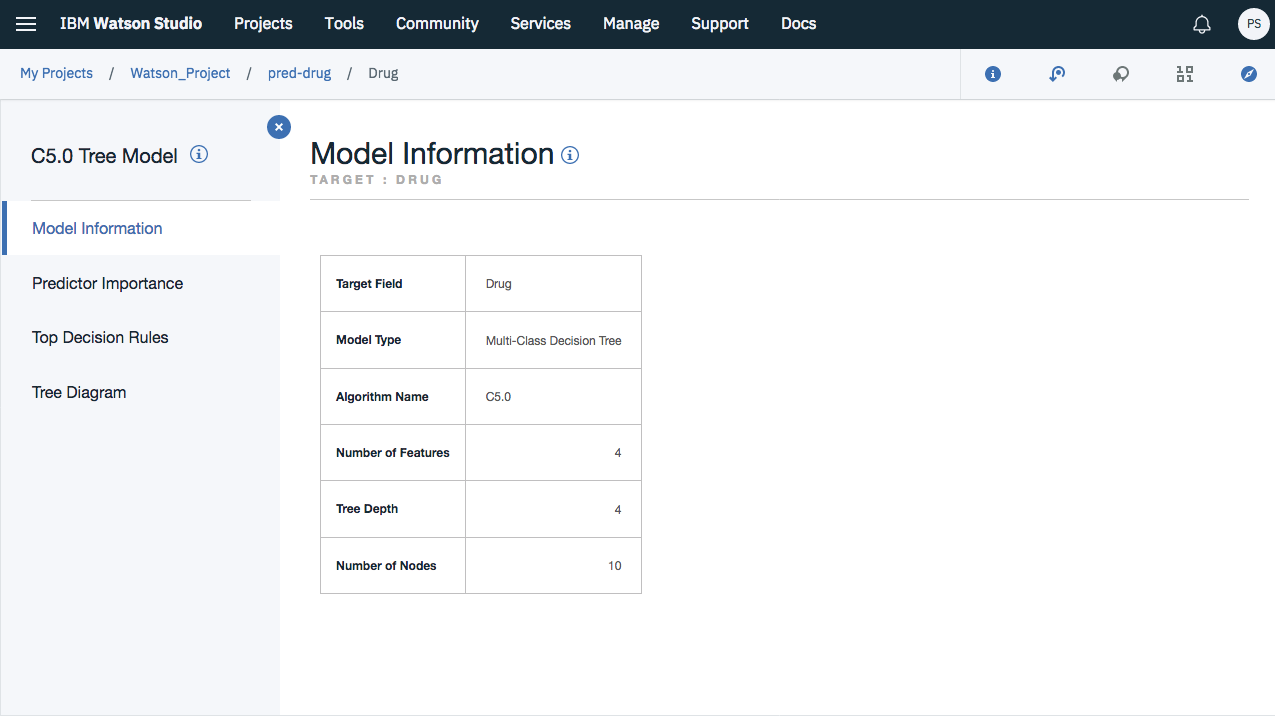


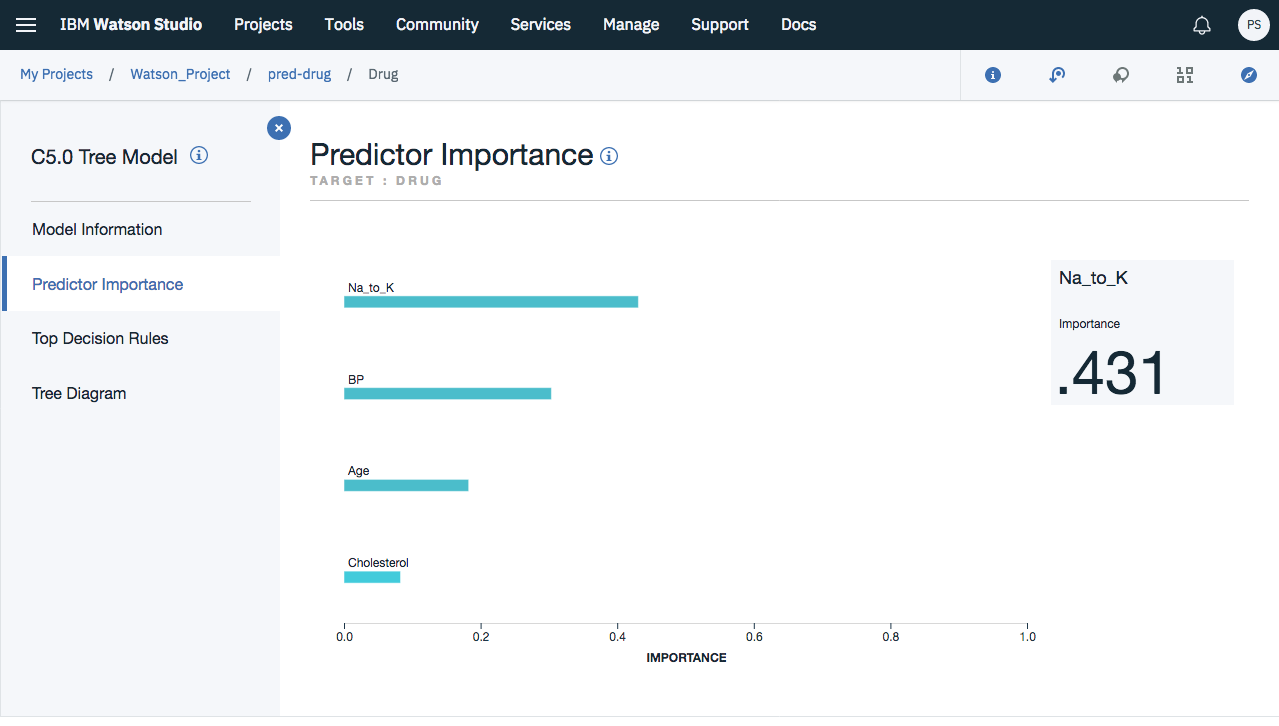
## 

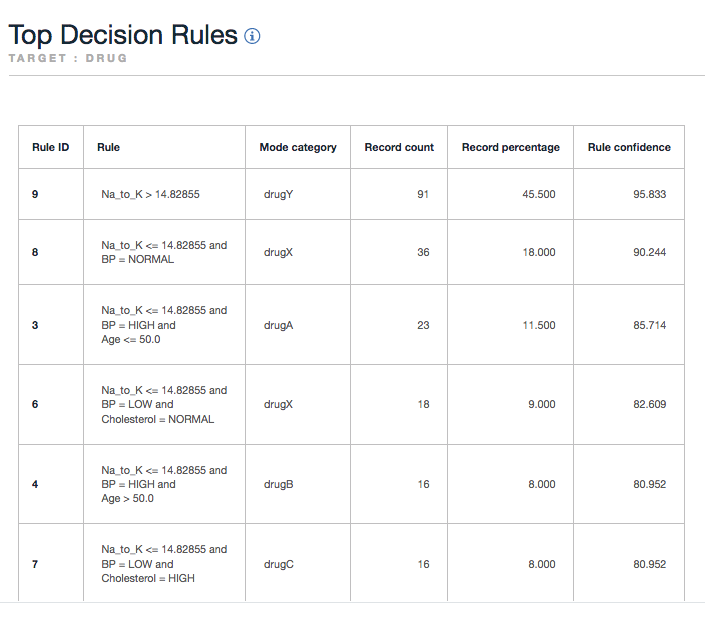
## **Building Model Using Decision Tree**



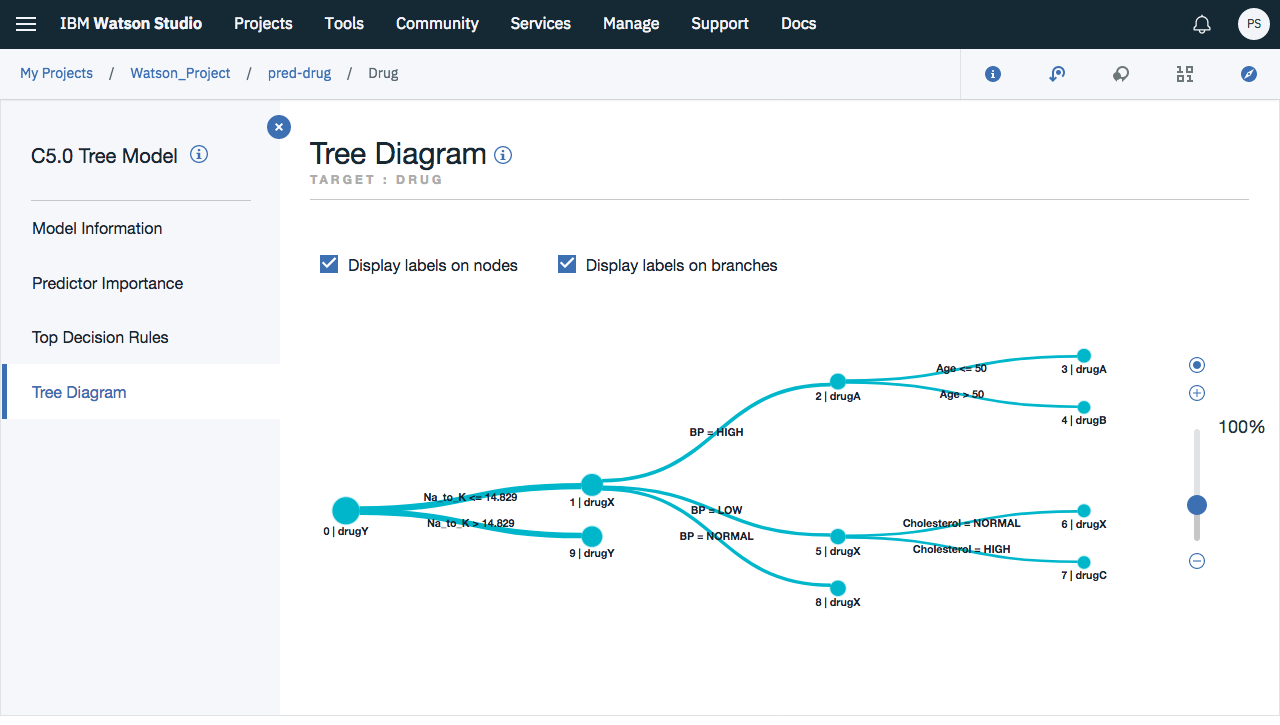






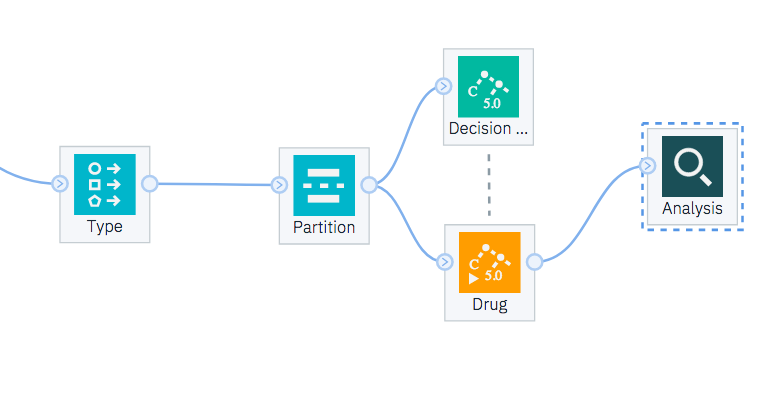


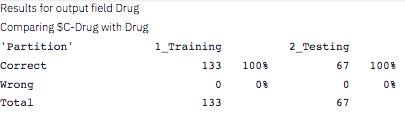
Now you can see the missing pieces of the puzzle. For people with a Na-to-K ratio less than 14.83 and high blood pressure, age determines the choice of drug. For people with low blood pressure, cholesterol level seems to be the best predictor.

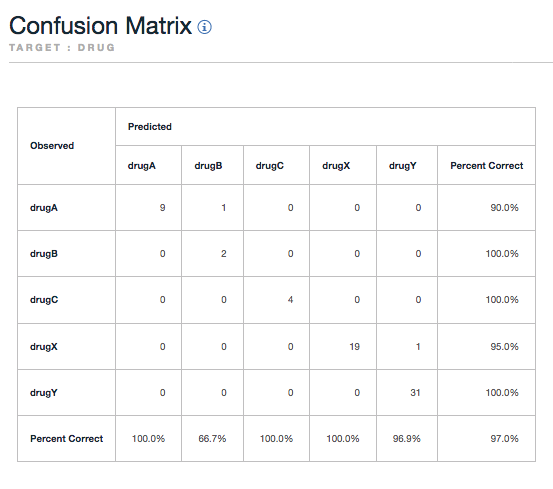


**Using an Analysis Node**

You can assess the accuracy of the model using an analysis node. Attach an Analysis node (from the Output node palette) to the model nugget, open the Analysis node and click Run.

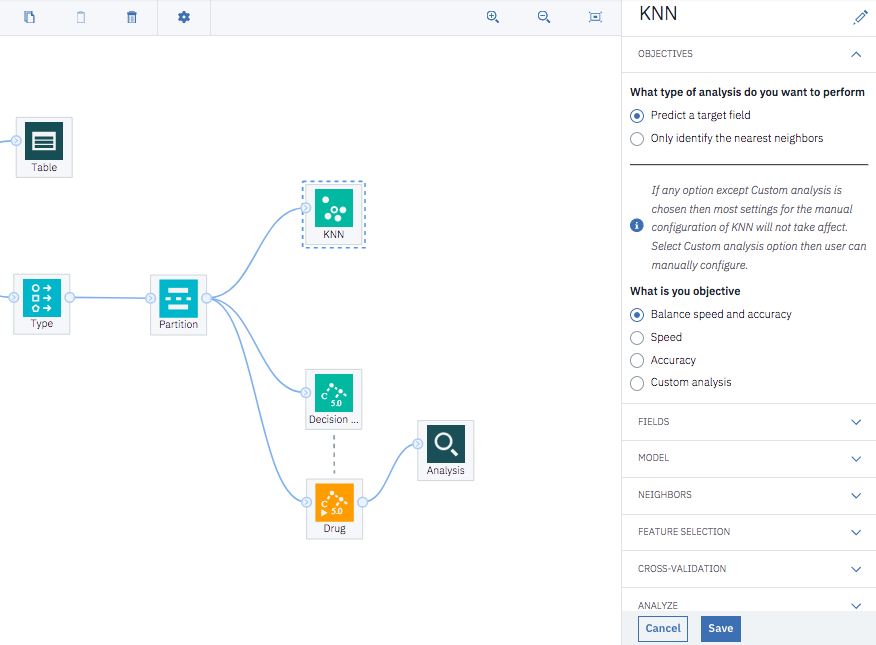


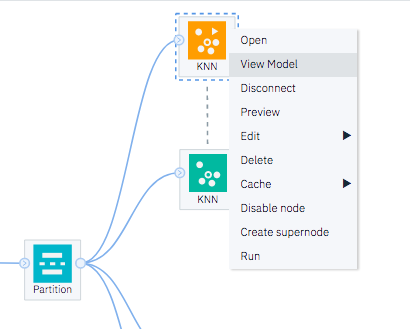


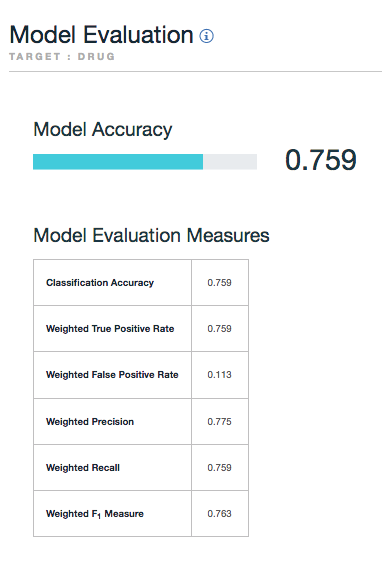


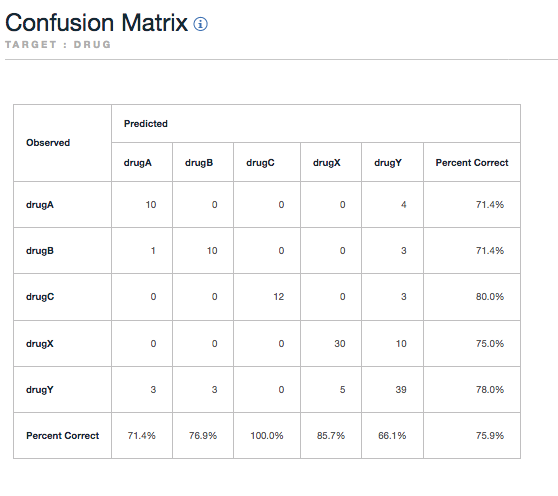
## 

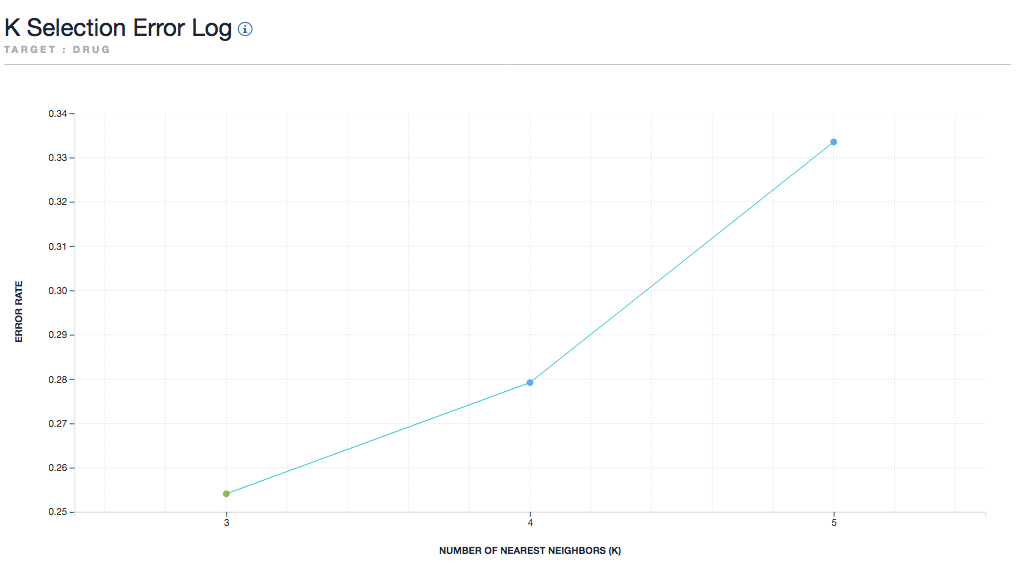
## **Building Model Using K-Nearest Neighbor**

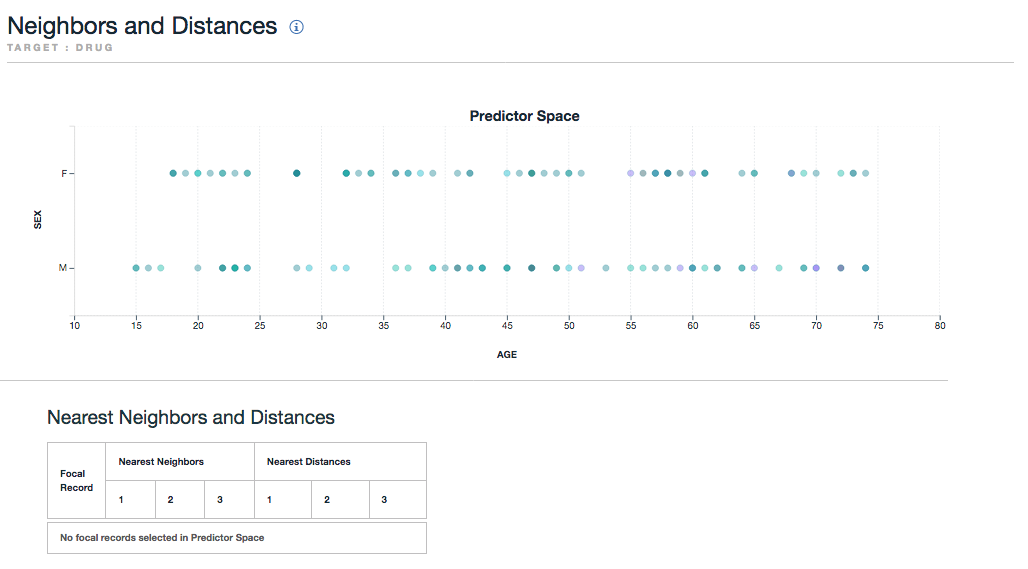








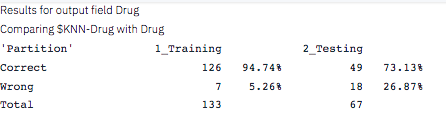




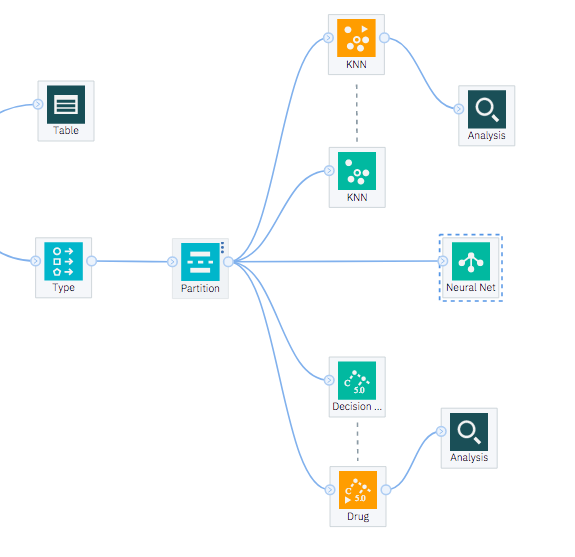
### **Using an Analysis Node**

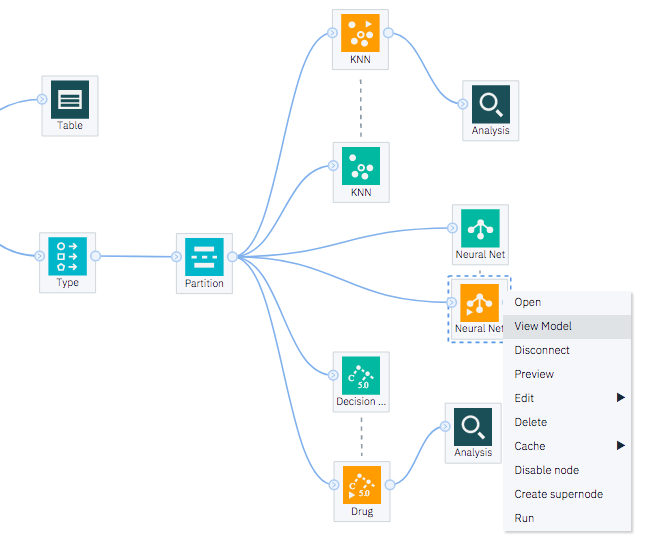
You can assess the accuracy of the model using an analysis node. Attach an Analysis node (from the Output node palette) to the model nugget, open the Analysis node and click Run.

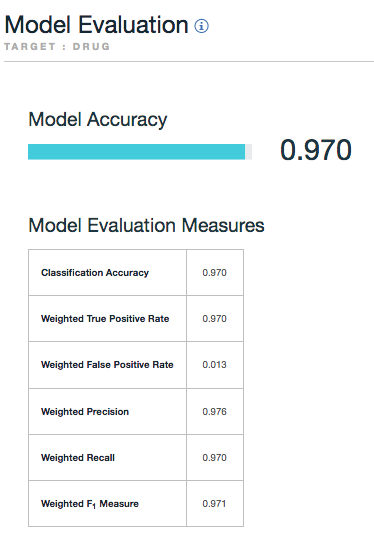
#### 

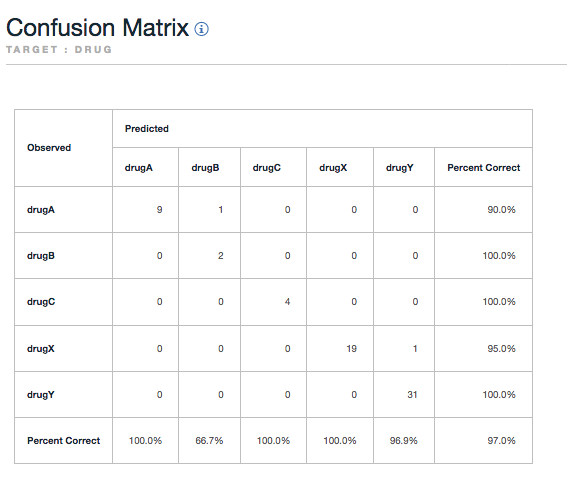


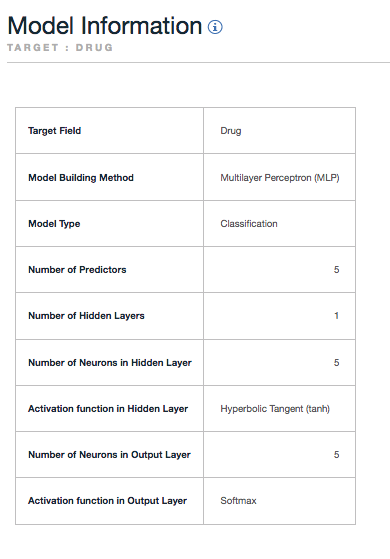
## **Building Model Using Neural Networks**

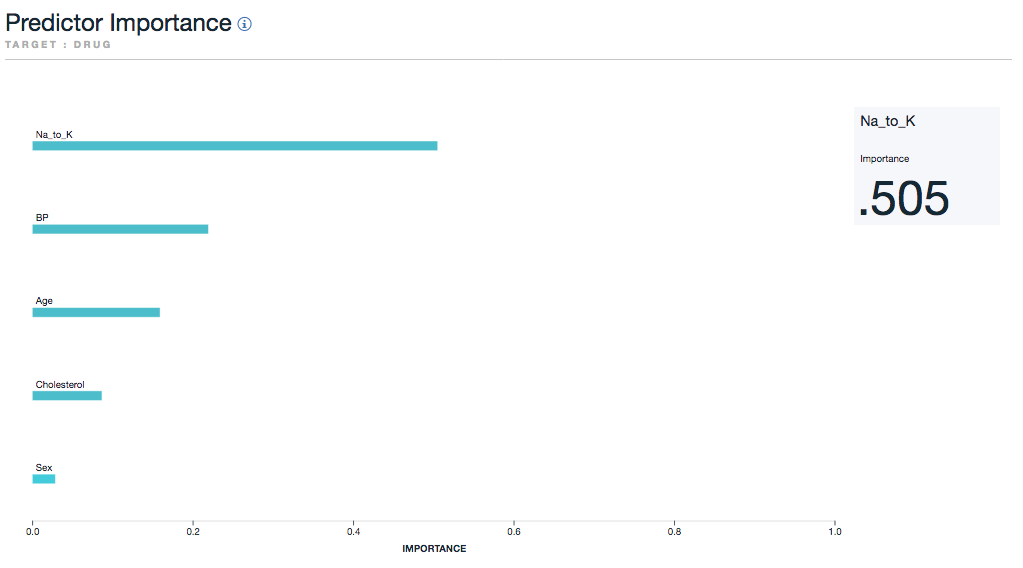


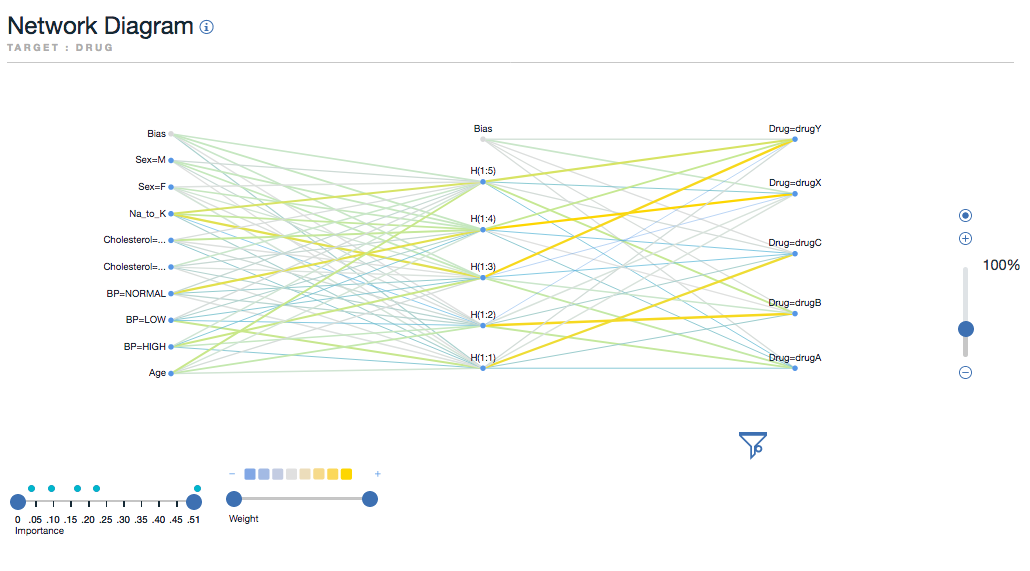






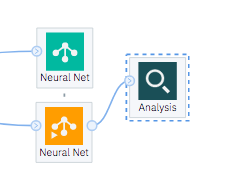


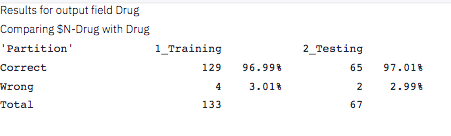




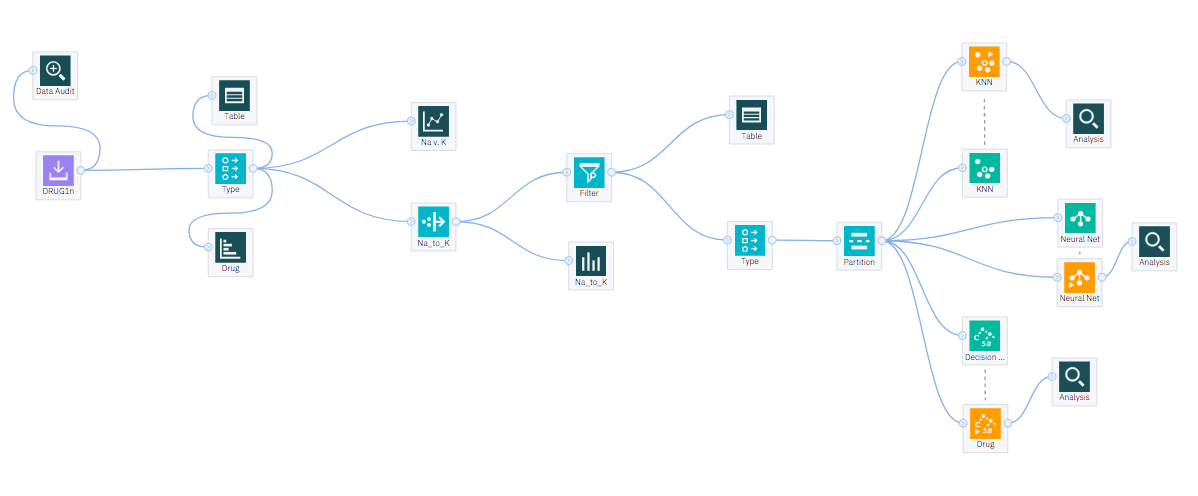
### **Using an Analysis Node**

You can assess the accuracy of the model using an analysis node. Attach an Analysis node (from the Output node palette) to the model nugget, open the Analysis node and click Run.



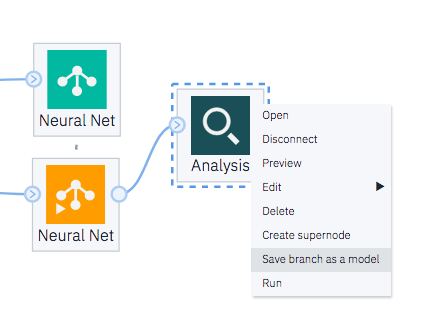


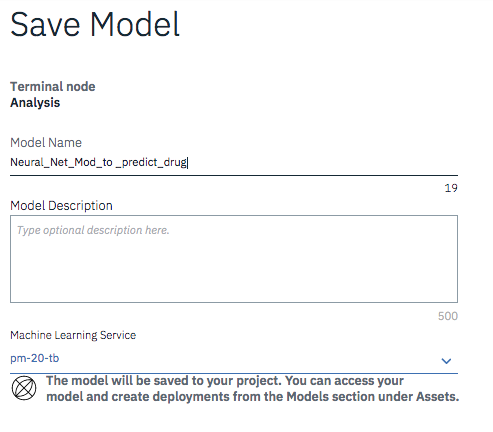
# Modeler Flow

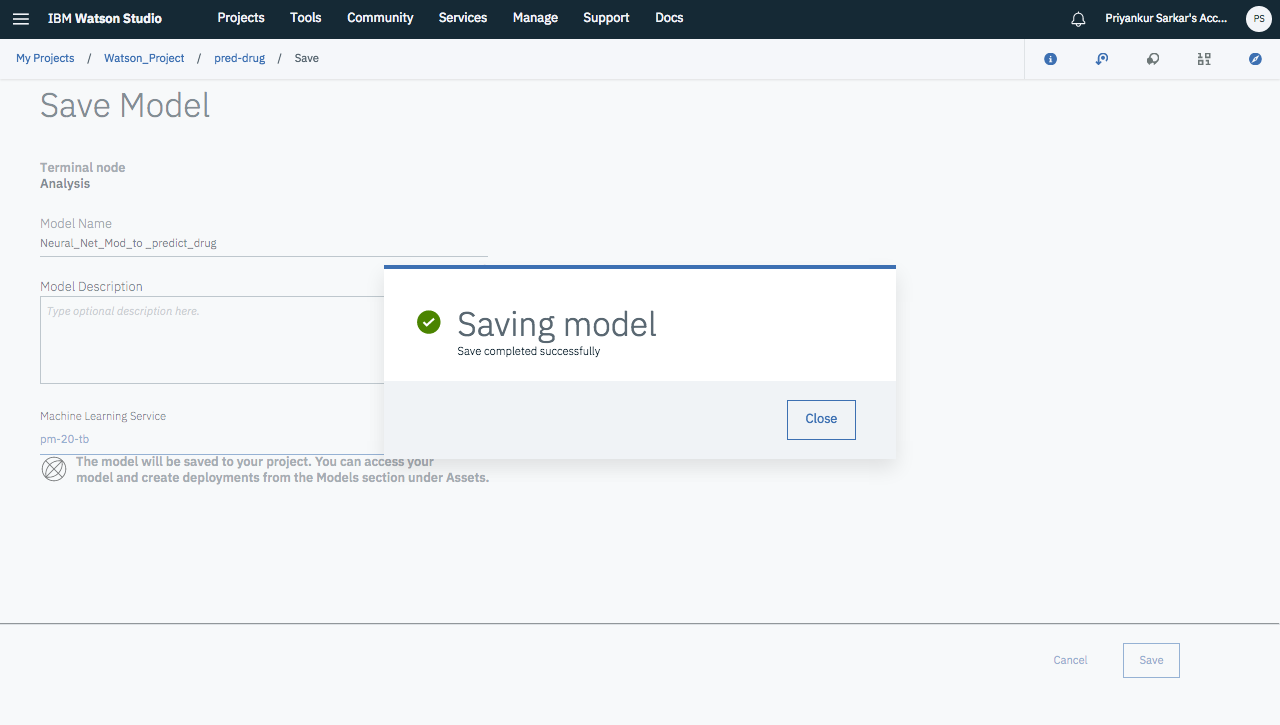


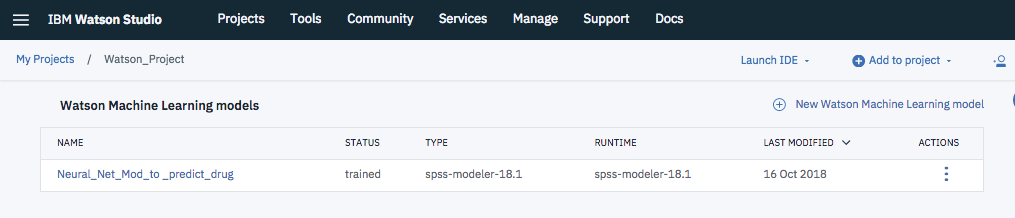
# Implementing the Model with Best Model Accuracy

As we have seen, the model built using Neural Network performs the best among the three and has a model accuracy of 97%. Let us implement the same model.

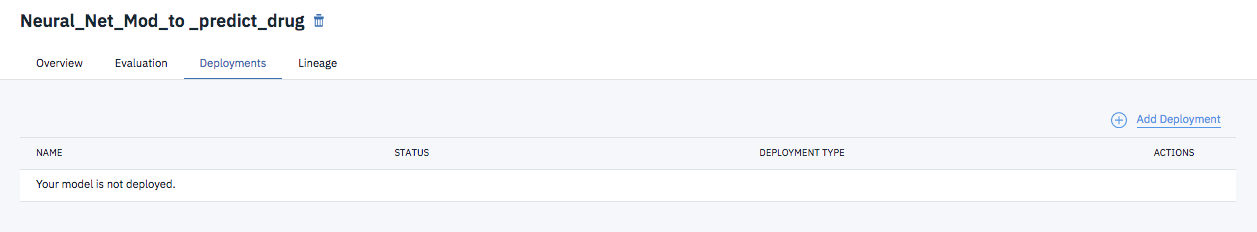


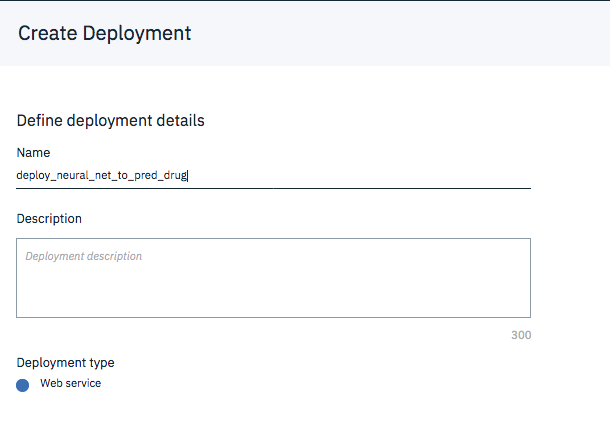


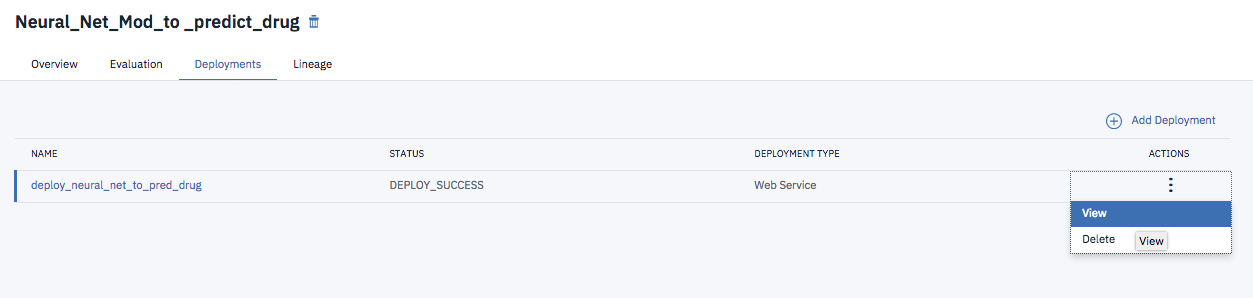




# Deployment







### **Test**



### **Implementation**

You can implement the model from implementation tab and can also check for the API specifications.

