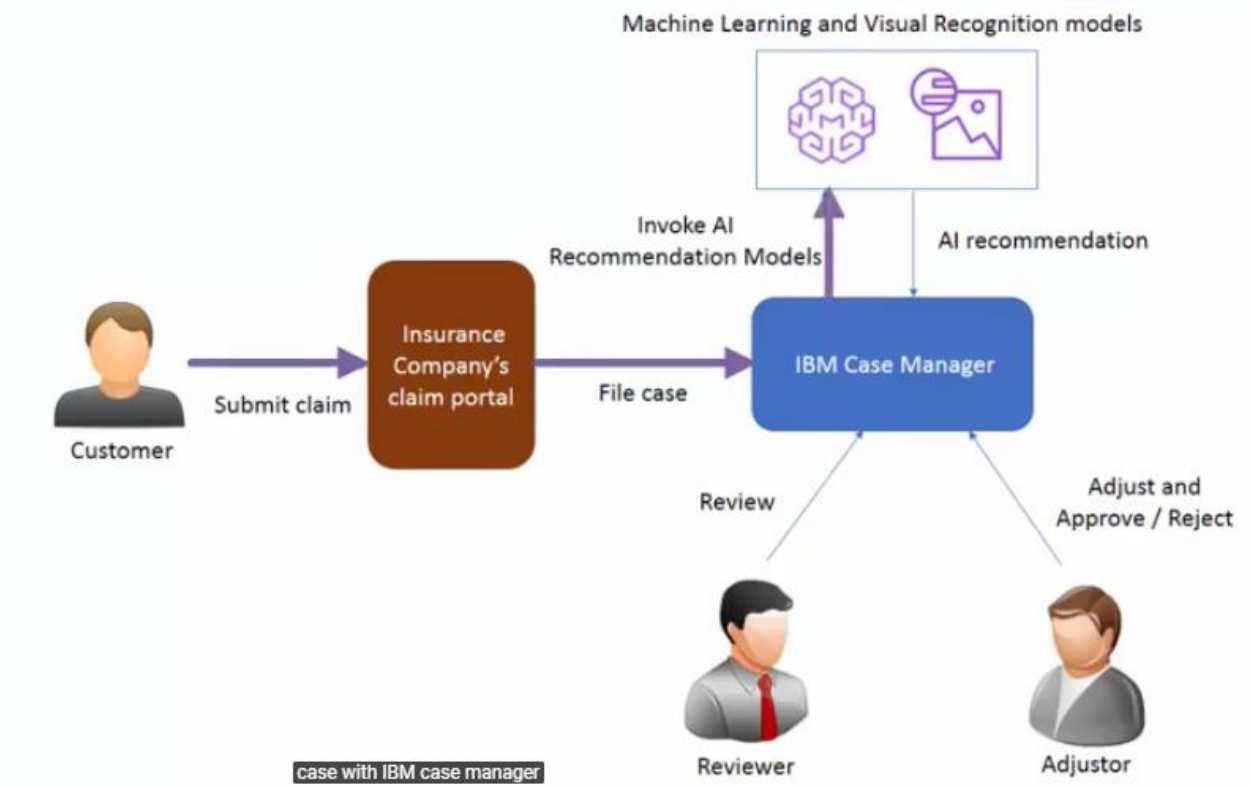
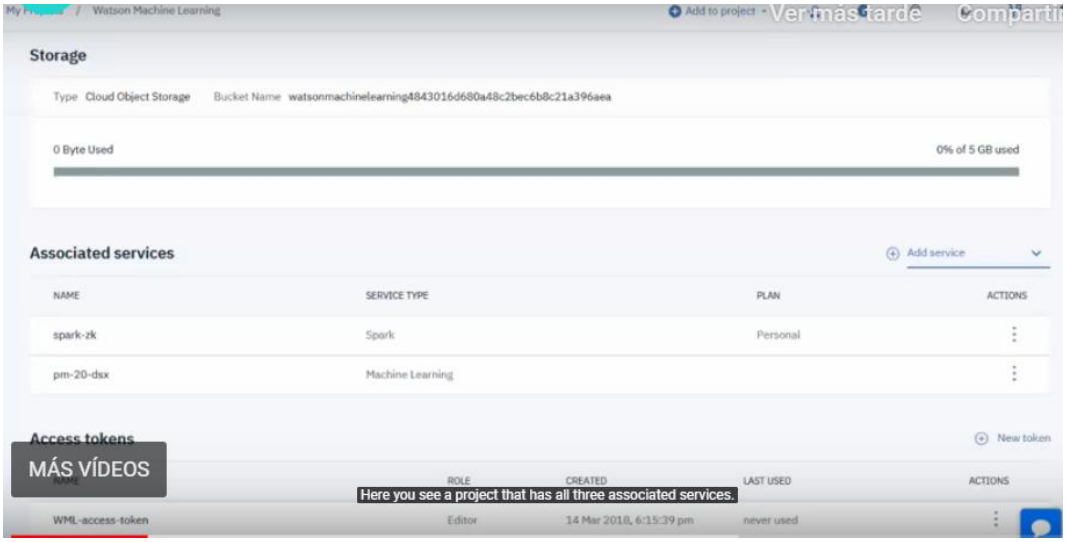
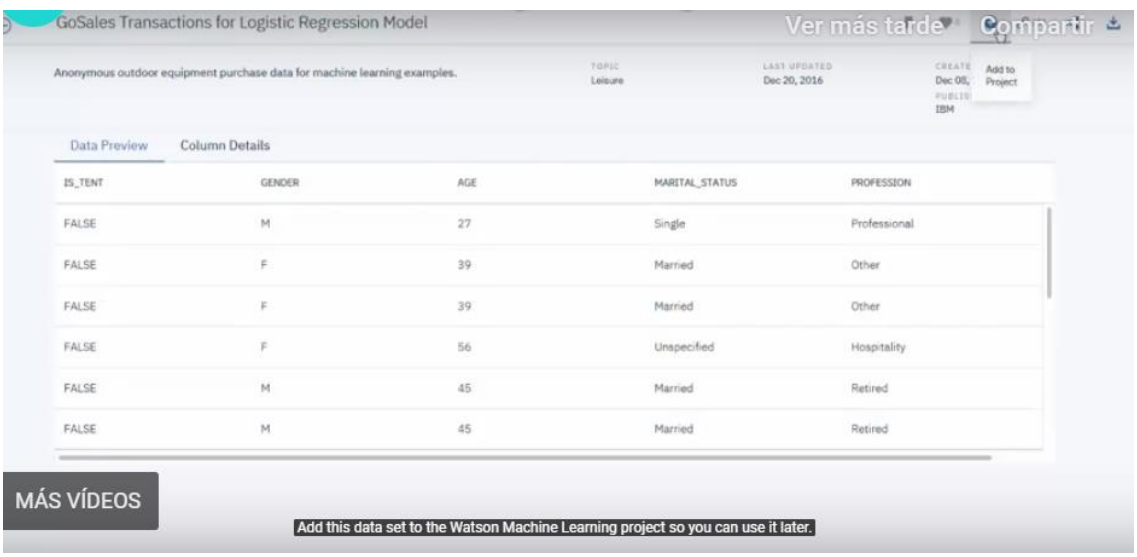
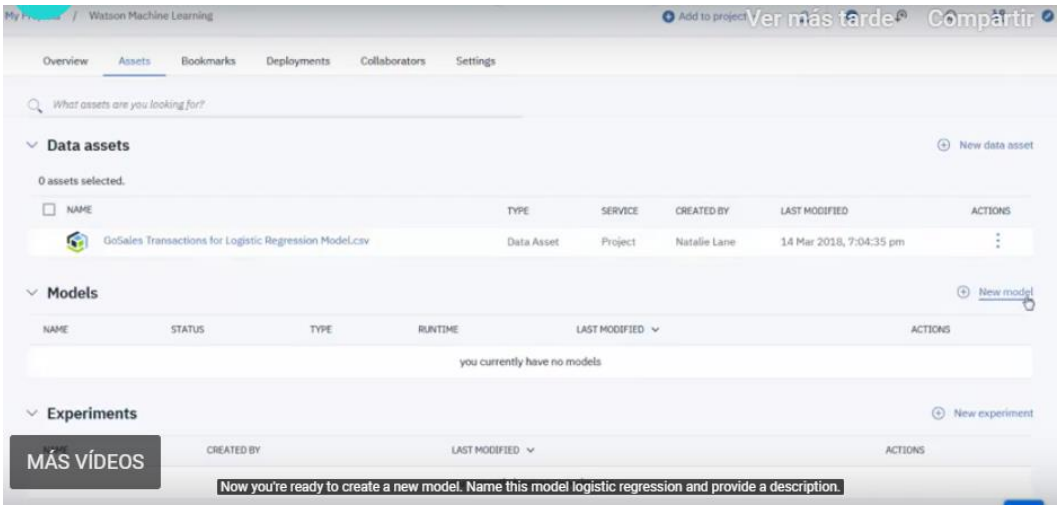


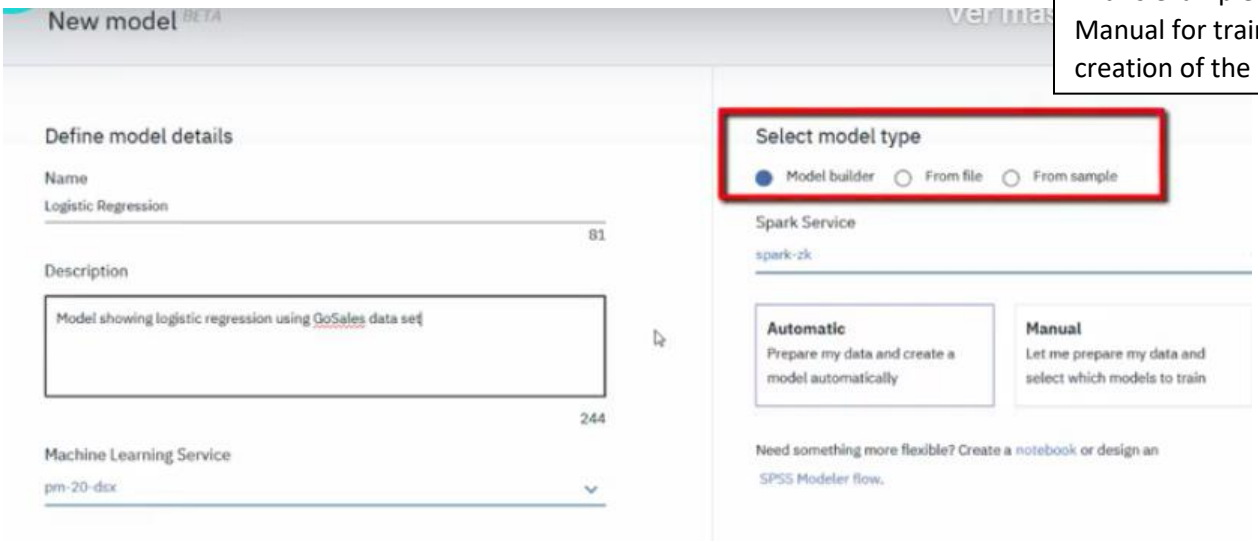

A project all three associated services

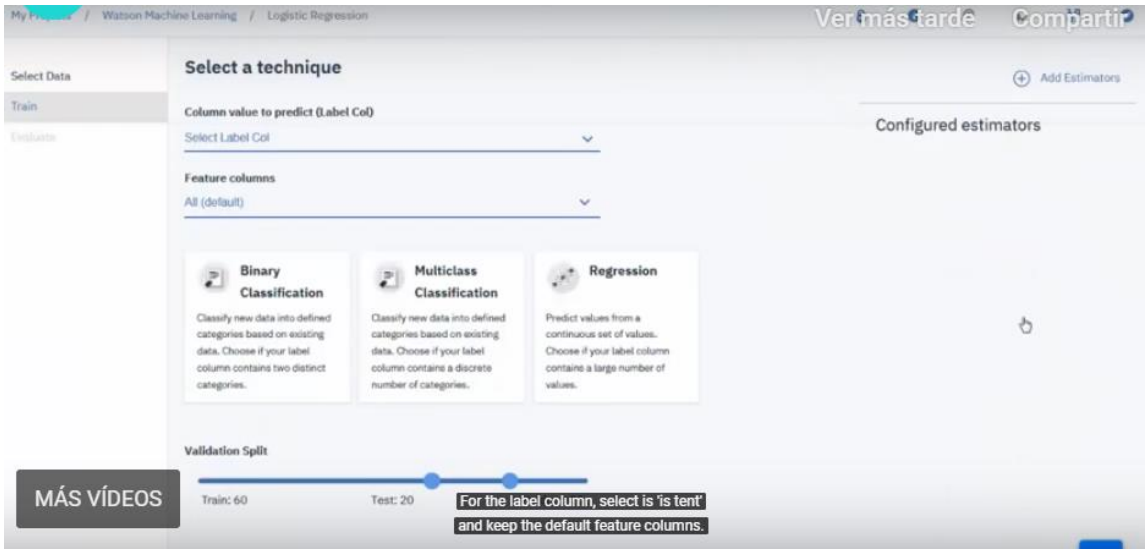



You can add the data set to Watson Machine Learning and you can see data preview and column details, in this case add data set of GoSales download since community IBM, late that upgrade the data set you can see in the project and specific in assets



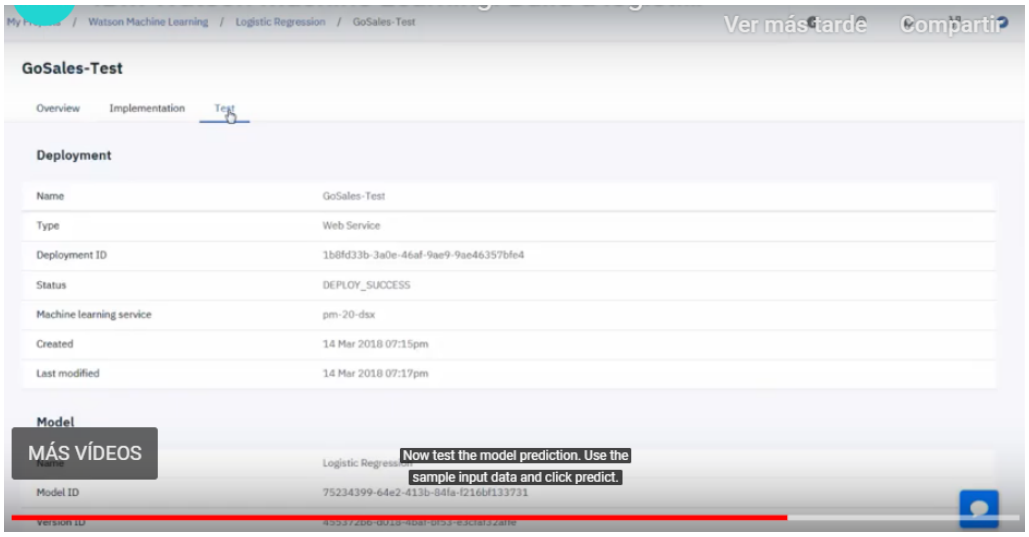

Now for create a new model, you can see that Machine Learning Service and Spark Service are predeterminate and you can create a model since beginning or charger a file with your code or charger a sample model. Also, in this example we choose the option Manual for training method in the creation of the model

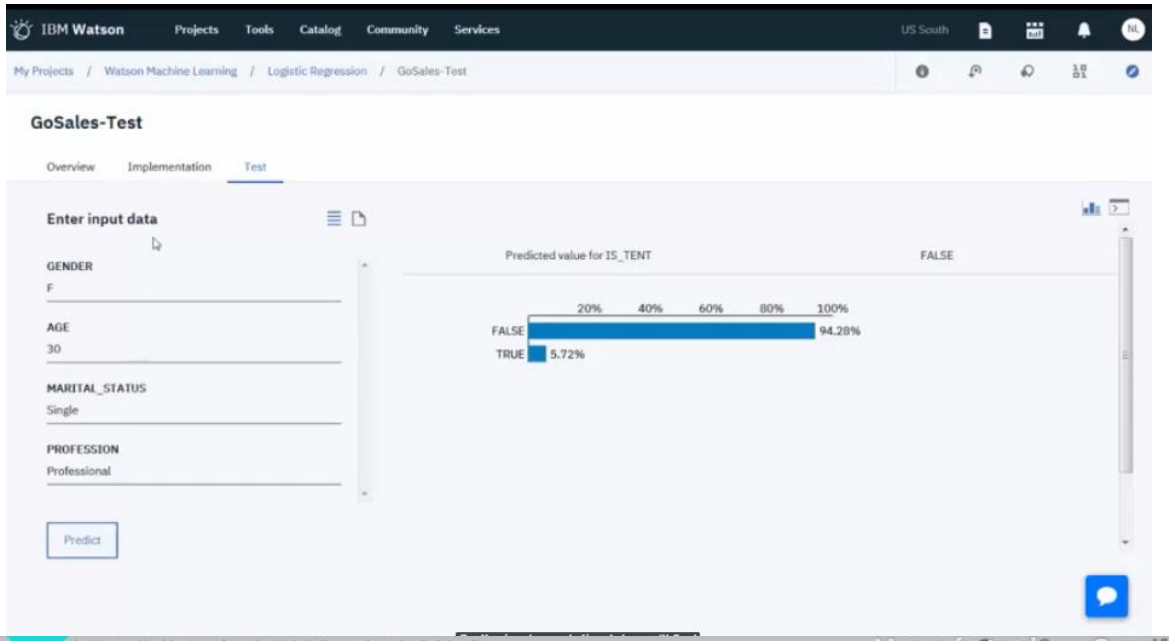


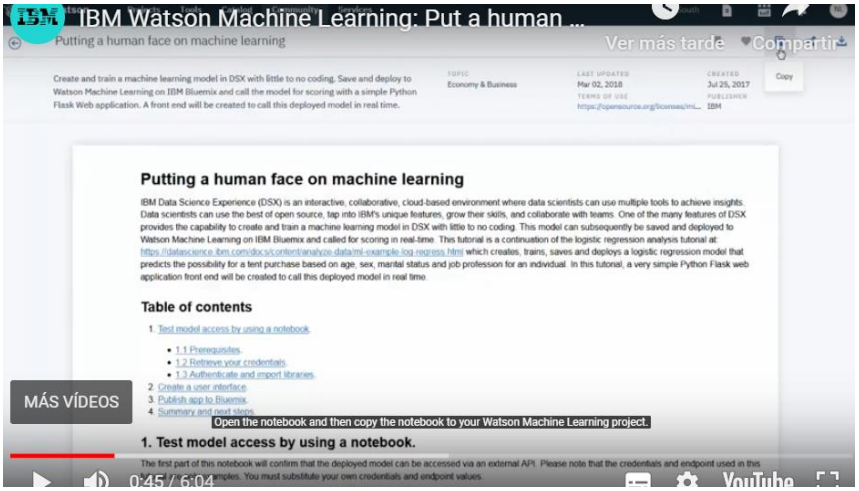

In this example we choose the column IS_TENT as label and keep feature columns default, late we selected a binary classification, and add an estimator, in this case we selected Logistic Regression, finally start the process and save the model


We added a deployment for our project, add a name for the deployment and save it, with the deployment save, late we view it and select the option TEST, in this case the deployment used the sample input data, so click in predict, we can change the variable for the prediction

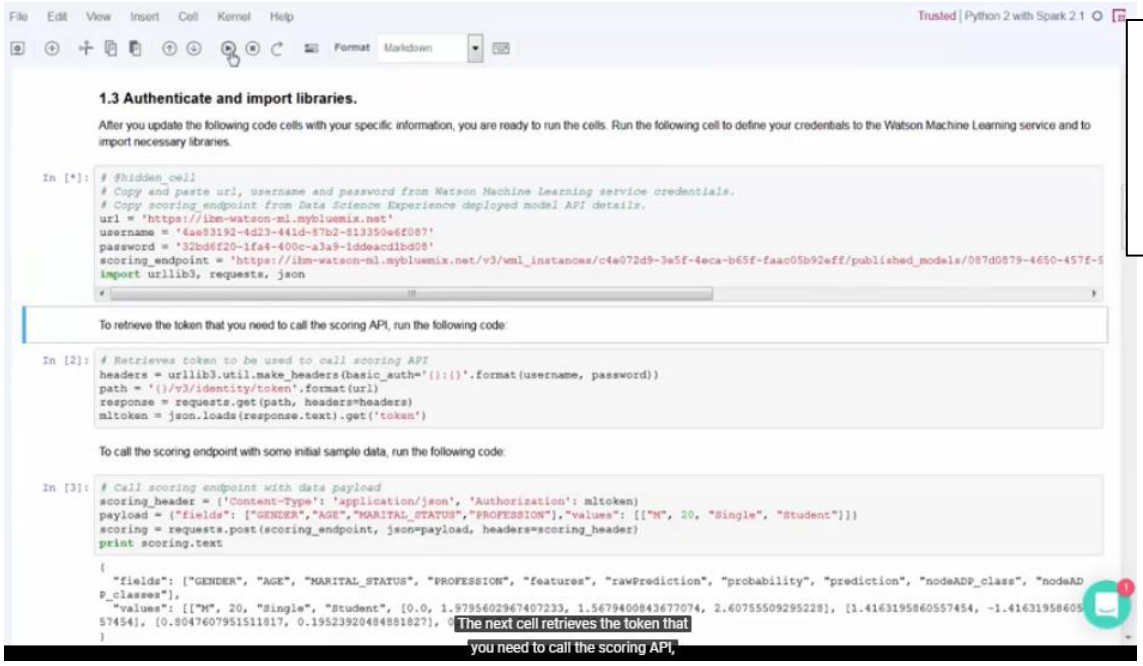


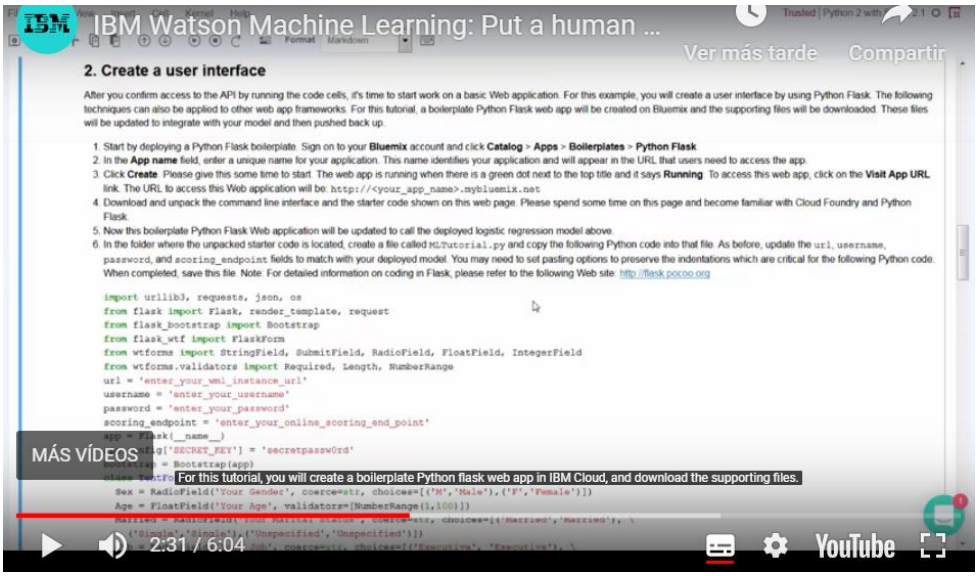

In the Implementation tab we can see the Scoring End-point for future predicts and we can find code snippets for implementation from your application, and finally we can view the API specification. This model is available on the assets tab in the project



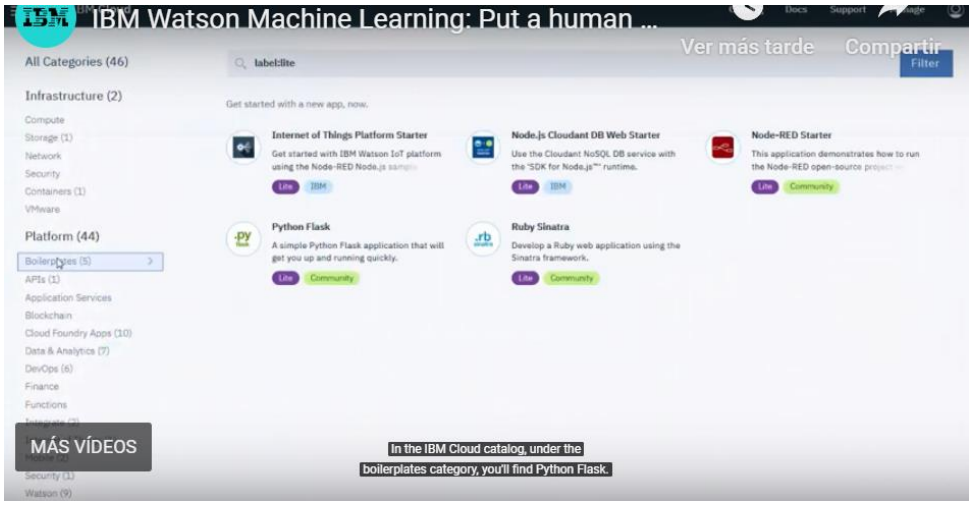

We copy the notebook in Watson Machine Learning project, the first step is to test access created previously in the logistic regression, the prerequisites to exception Python Flask are implement in Cloud, for the service credentials, we find the URL username and password in the service credentials tab and you need also the scoring end-point of your test logistic regression model



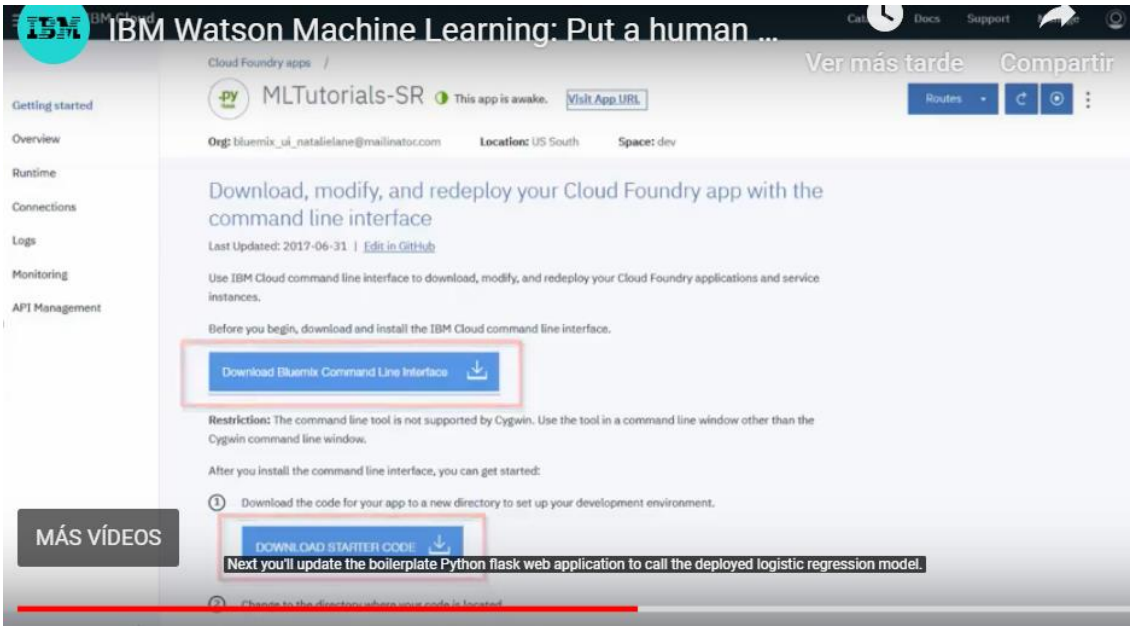

In the first cell we upload our data, in the second cell only is for check the view of our API, and in the third cell we call the scoring end-point with someone initial sample data



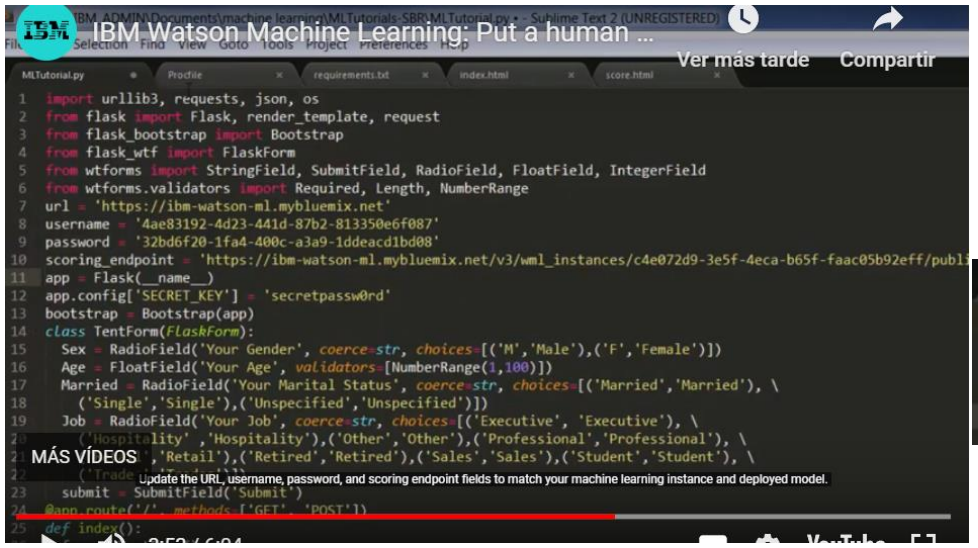

For this example, we'll create a boilerplate Python Flask web app in IBM Cloud and download the supporting files, you'll update the files to integrate with your model and then push it back up

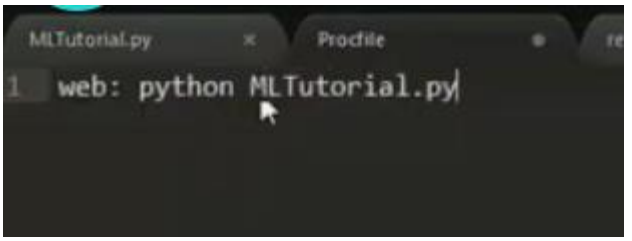


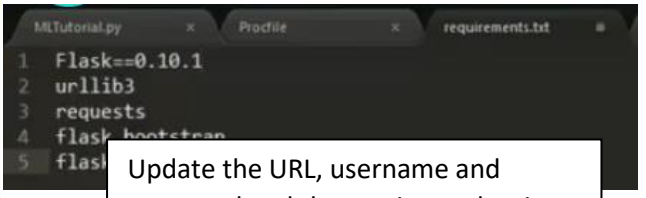

In the IBM Cloud catalog, we find boilerplates and we selected Python Flask, here created it, we named it



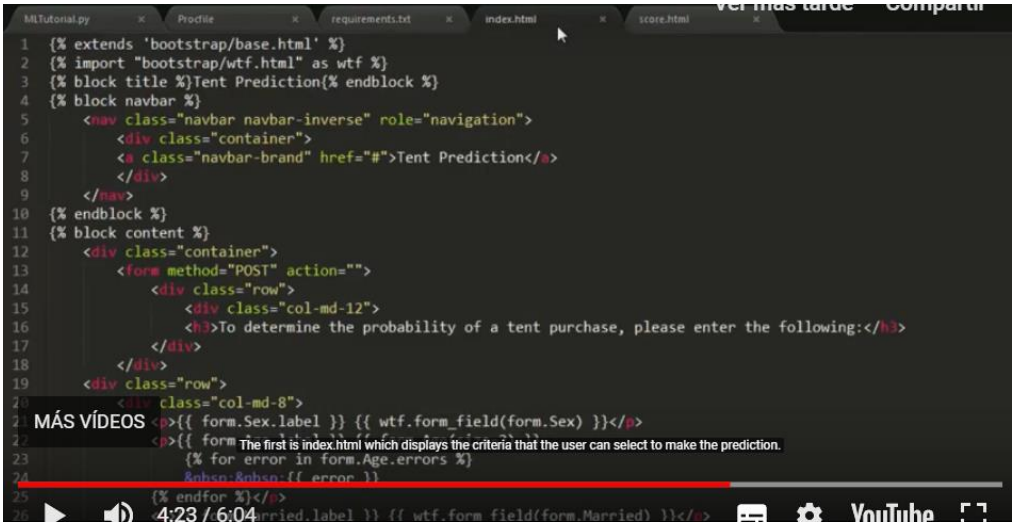

A time creates the python flask we can see the URL of app and late we downloaded command line interface and starter code, next you'll update the boilerplate python flask, in the folder when you unpack the starter code, created a file called mltutorial.py and copy the python code from this notebook into that file

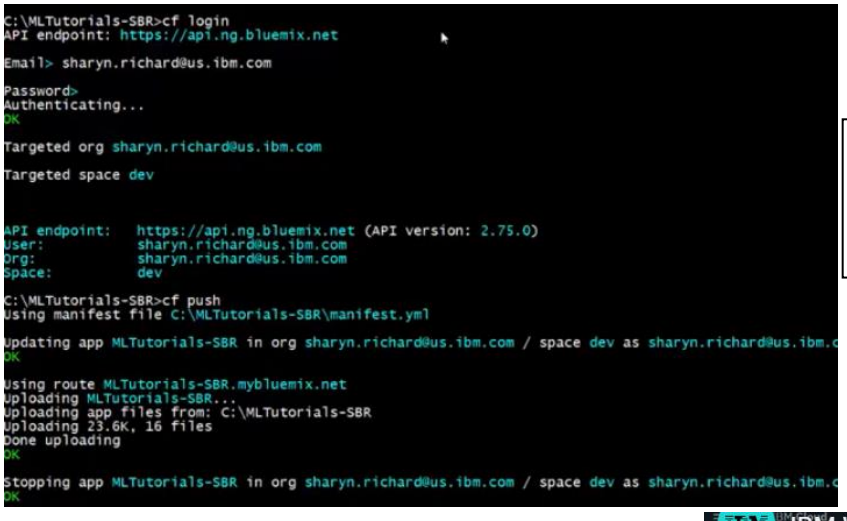


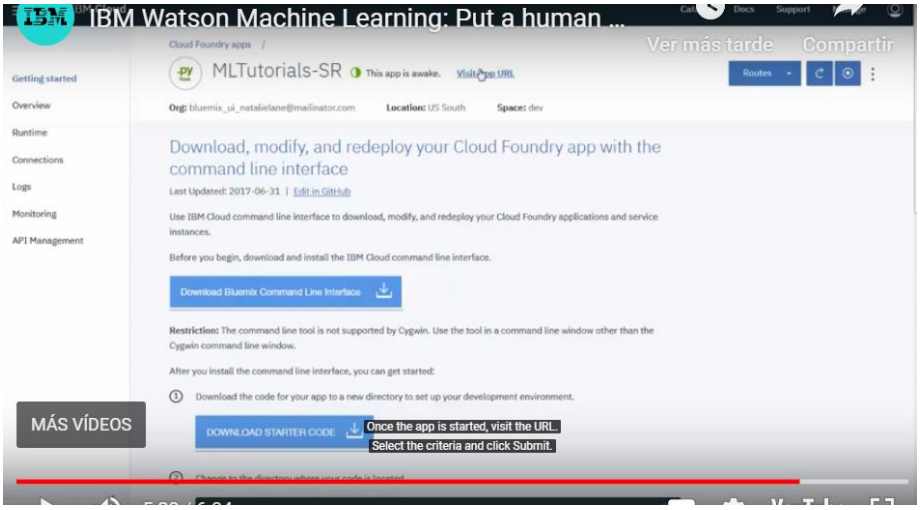

Update the URL, username and password and the scoring end-point, saved the file, next is edit procfile and change the python file name for Mltutorial.py, late edited the requirements.txt file and paste in the code from the notebook. Now you'll add two files to the template's subfolder, the first is index.html can use for the prediction and the second is score.html displays the results of the prediction

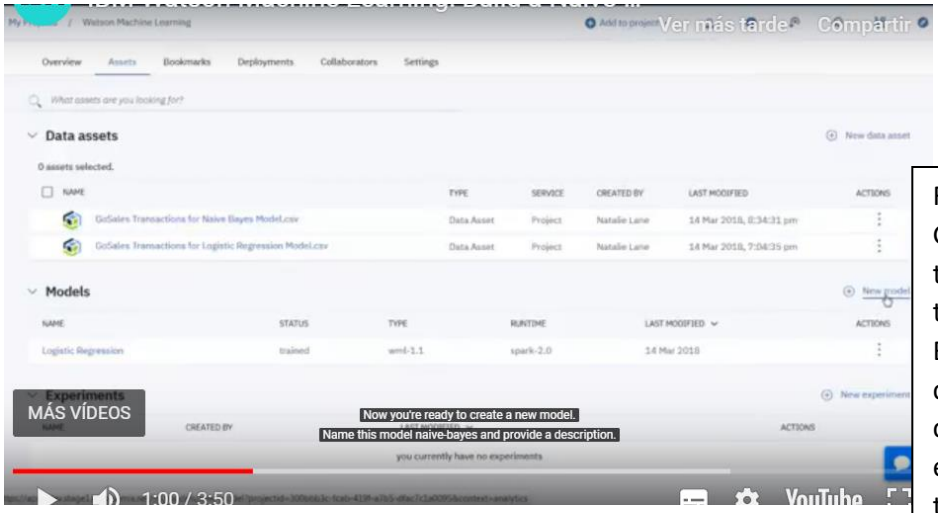


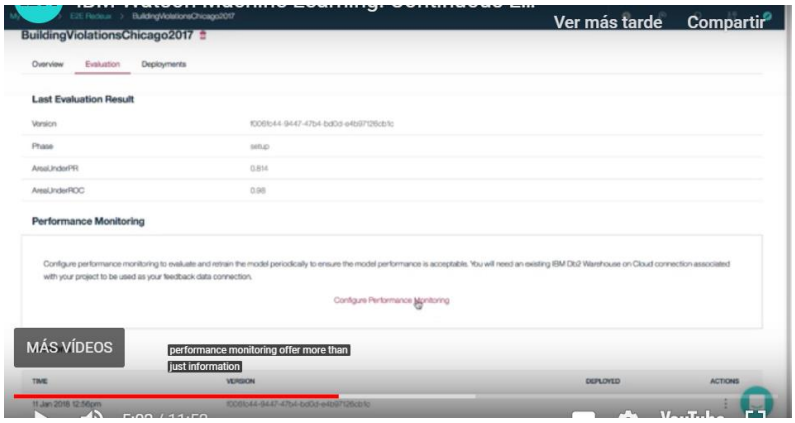

For run the app, we open the file and cf login, here input the email and password late cf push

Once the app started, we visited the URL and we can check the results

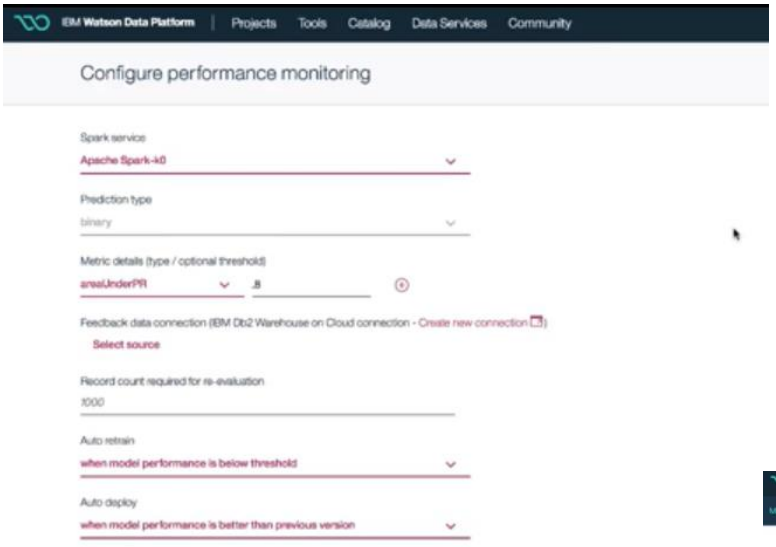


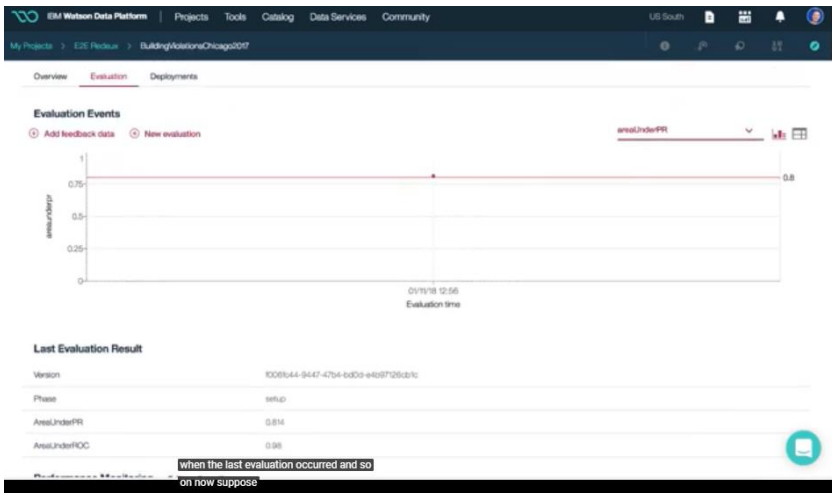


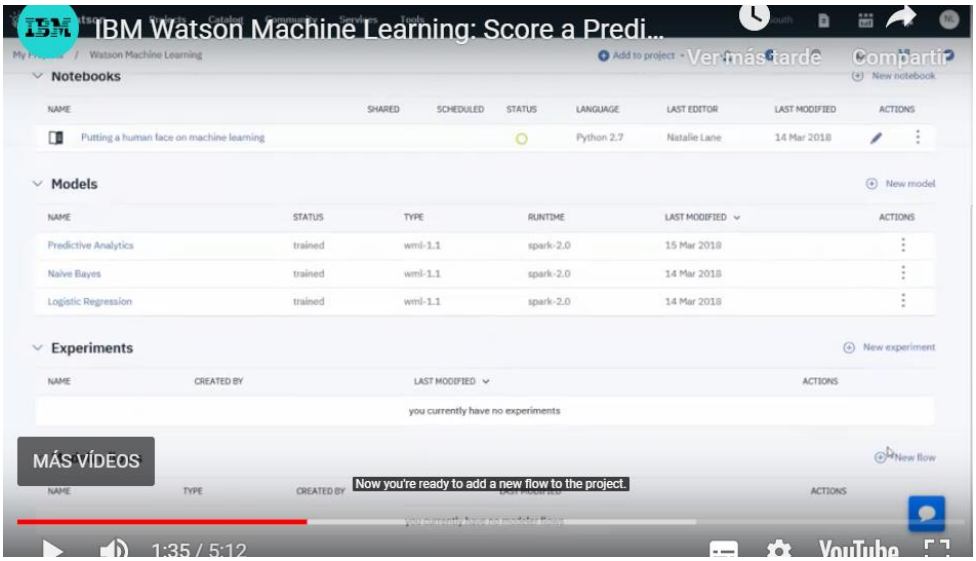

For the Naive-Bayes first add the asset with the name GoSales for Naive-Bayes, late create a new model, in this case created a new model of the way Manual, for the training used the data set GoSales For Naïve-Bayes and in the column value predict used the column PRODUCT_LINE, we used default in feature columns and we selected multiclass classification in estimator selected Naïve Bayes, late train and save the model. In the deployment tab add a new deployment late viewed the deployment



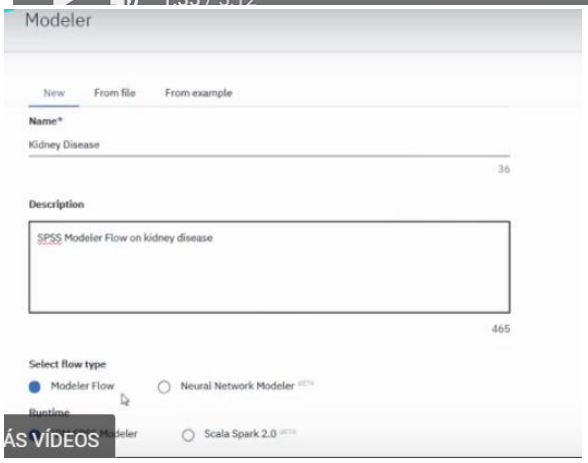

Once created the service into data services, add D2 Warehouse is necessary create the model in this case a logistic regression, for creates this, follow the same steps as you use only Watson Machine Learning, in this example create with two estimators, logistic regression and decision tree. Late in Evaluation tab click in Configure Performance Monitoring, with this option we can see configuration advancer for the evaluation, for example in this case conditional the ROC curve when down over time so retrain the model with new data. In create new connection we put the data base contains the information for our model, and for refresh the information with new data we selected the option SELECT SOURCE and we chose the data base that is refresh, as next step we can charger new data for example in this case the information was about of September but once terminated the period we charged the info of October and the successive. In the DEPLOYMENTS tab we can create a API for do predictions in real time

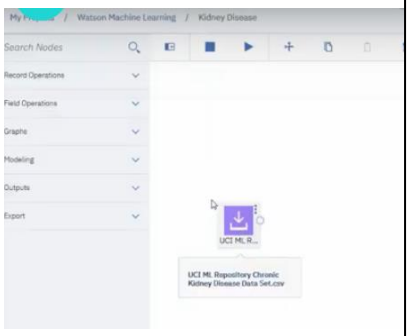


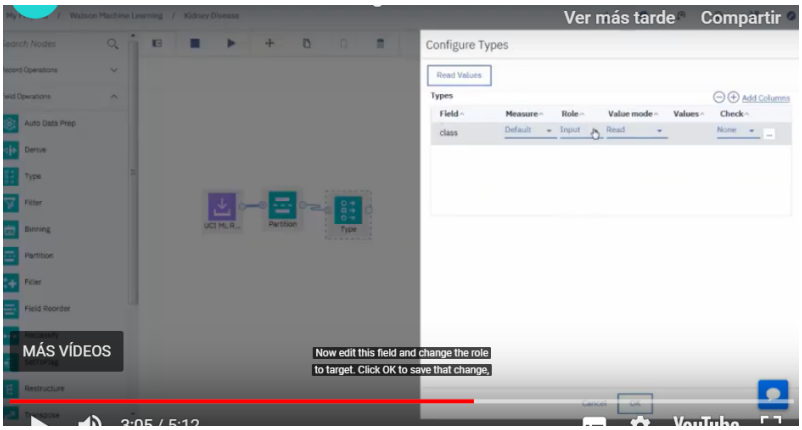



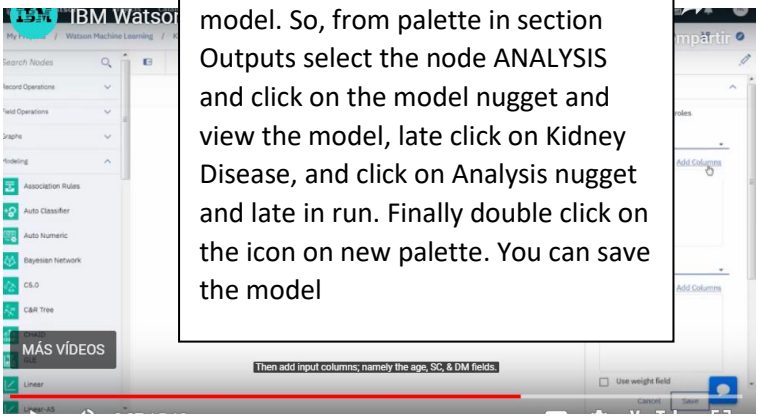


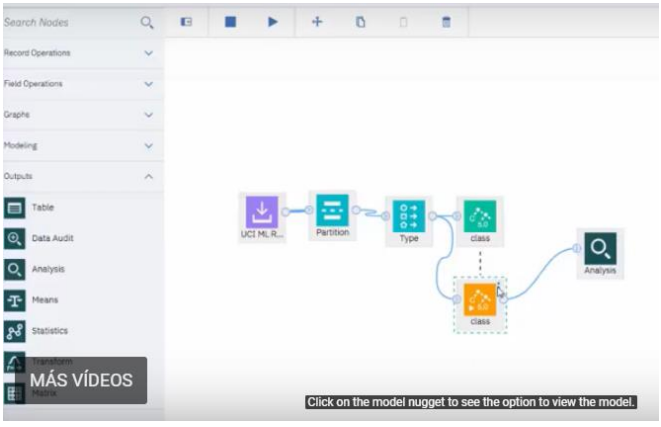

In this example we used the data set chronical kidney disease, we created new project with three services used and late add a new flow, in this flow selected modeler flow and SPSS IBM Modeler. Once create the model, open for setting our model, here there are our data set, selected the chronical kidney disease. The second node was partition, this for training and test our model. The third node was TYPE, this for create a target variable, in this case CLASS, save the changes. Back on the palette, in the modeling section we chose C5.0, here we selected the field roles CLASS and added columns, the columns are the variable for help to classifier the target value, in this case Age, SC and dm, save the changes and run the model with the icon PLAY. Will appear a new node this is trained the model. So, from palette in section Outputs select the node ANALYSIS and click on the model nugget and view the model, late click on Kidney Disease, and click on Analysis nugget and late in run. Finally double click on the icon on new palette. You can save the model

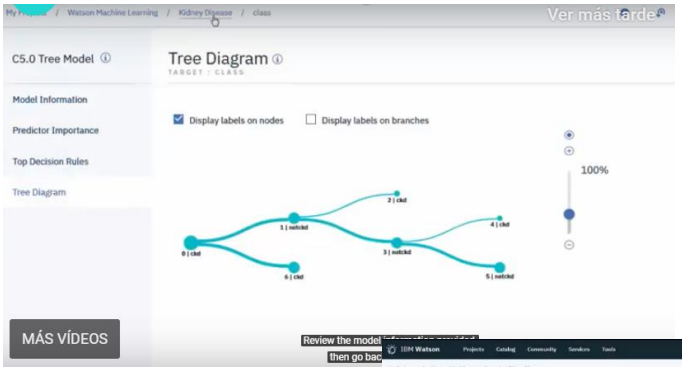


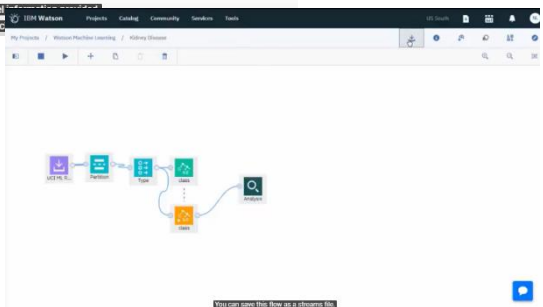





- w.oosterbosch@nl.ibm.com
  - Natasha.Savic@ibm.com