

databricks 02.1 - Machine Learning - Inference

(<https://databricks.com>)

Churn Prediction Inference - Batch or serverless real-time

After running AutoML we saved our best model our MLflow registry.

All we need to do now is use this model to run Inferences. A simple solution is to share the model name to our Data Engineering team and they'll be able to call this model within the pipeline they maintained.

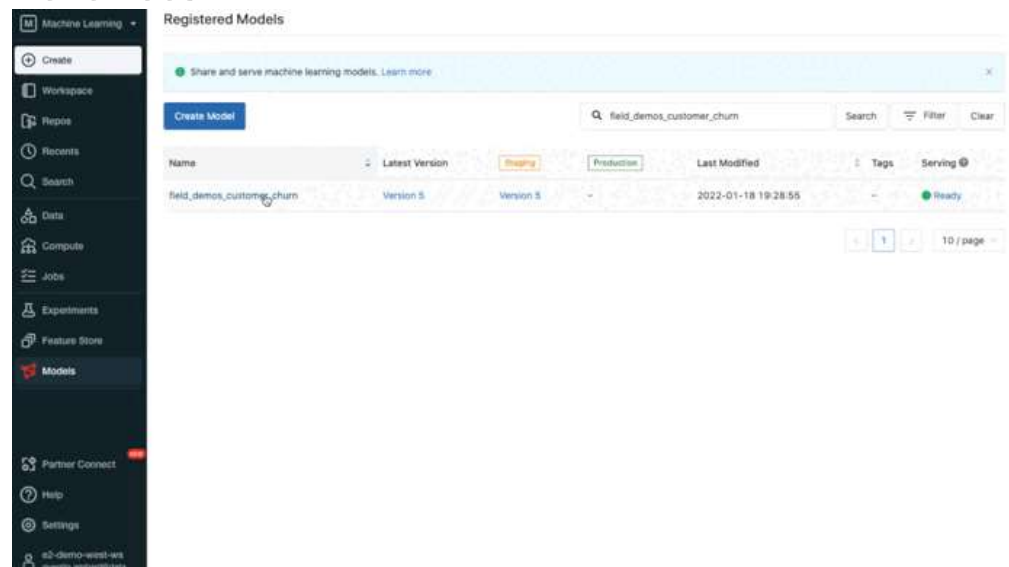
This can be done as part of a DLT pipeline or a Workflow in a separate job. Here is an example to show you how MLflow can be directly used to retrieve the model and run inferences.

Deploying the model for batch inferences

Now that our model is available in the Registry, we can load it to compute our inferences and save them in a table to start building dashboards.

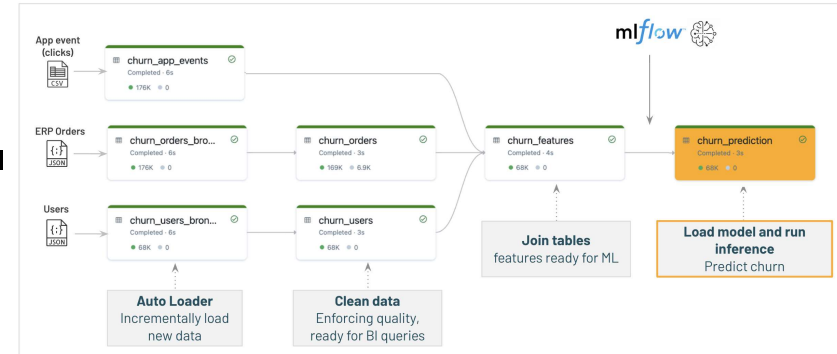
We will use MLflow function to load a pyspark UDF and distribute our inference in the entire cluster. If the data is small, we can also load the model with plain python and use a pandas Dataframe.

If you don't know how to start, Databricks can generate a batch inference notebook in just one click from the model registry: Open MLflow model registry and click the "User model for inference" button!



Our Data scientist team has build a churn prediction model using Auto ML and saved it into Databricks Model registry.

Note that we don't have to worry about the model framework (sklearn or other), MLflow abstracts all that for us.



Loading the model

```
import mlflow
```

#		Stage/version		
#	Model name			output
#				
modelURL = "models/" + modelName + "/Production"			#	
print("Retrieving model " + modelURL)			#	
predict_churn_udf = mlflow.pyfunc.spark_udf(spark, modelURL, "int")				
#We can use the function in SQL				
spark.udf.register("predict_churn", predict_churn_udf)				

Creating the final table

```
model_features = predict_churn_udf.metadata.get_input_schema().input_names()
predictions = spark.table('churn_features').withColumn('churn_prediction', predict_churn_udf(*model_features))
predictions.createOrReplaceTempView("v_churn_prediction")
```

```
%sql  
create or replace table churn_prediction as select * from v_churn_prediction
```

```
%sql  
select * from churn_prediction
```

Next up

Explore the data with SQL and create visualisations ([\\$.03%20-%20BI%20and%20Data%20Warehousing](#))

