Overview

TODO: Short Introduction, Prerequisites etc.

Use-Case

You are a data scientist employed by ABCOnlineRetailXYZ Ltd. The online retailer has not been doing that great lately and has been losing customers to competition. While it is clear that its customer churn rates have been growing, figuring out which customers are more likely to churn is proving hard.

Your manager came by your desk today, gave you a couple of files containing some customer-related data and asked you to build a model helping the business understand which of its customers may churn in the near future, so that Marketing could proactively target them with personalized incentives and discount vouchers.

You have a lot of other work to do and are a short on time,and are in need of inspiration – your Python skills are a bit rusty.  
You are aware that IBM provides market-leading data science tooling and decide to use IBM’s Watson Studio running on CPDaaS to build and test your model.

Graphical user interface, application

Description automatically generated

Figure 1:IBM’s market position

IBM’s Cloud Pak for Data provides a range of capabilities catering for the whole data science life cycle – from data preparation, to model build, deployment, training and monitoring.

Graphical user interface, application

Description automatically generated

Figure 2: IBM's Data Science Portfolio

The platform’s Data Science and ML component(s) sit under the Watson Studio branding umbrella. Cloud Pak for Data’s Watson Studio services (Watson Studio / Watson Studio Build, Watson Machine Learning / Watson Studio Deploy, Watson Openscale / Watson Studio Trust) enable the Build, Run and Manage capabilities of the MLOps lifecycle.

Graphical user interface

Description automatically generated with medium confidence

Figure 3: Watson Studio's components

In this Lab, you will:

* [Use AutoAI to automatically build the most optimal customer churn prediction model based on your data sets](#bookmark)
* [And, finally, deploy and test that model with Watson Machine Learning (Watson Studio Deploy capabilities).](#bookmark1)

… and, your model will be built (for you!) and deployed in minutes - not hours or days!

Building Models with AutoAI

In this section, we are going to build a machine learning model using [AutoAI](https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=models-autoai). The AutoAI graphical tool in Watson Studio automatically analyses your data and generates candidate model pipelines customized for your predictive modelling problem.

These model pipelines are created iteratively as AutoAI analyses your dataset and discovers data transformations, algorithms, and parameter settings that work best for your problem setting.  Results are displayed on a leaderboard, showing the automatically generated model pipelines ranked according to your problem optimisation objective.

Timeline

Description automatically generated

Click the Add to New Assets button and add a new AutoAI experiment.

Graphical user interface, application

Description automatically generated

Give it a name – e.g. Churn prediction, click Create to move to the next step.

Graphical user interface, text, application

Description automatically generated

Choose Select from Project on the next screen.

Graphical user interface, text, application

Description automatically generated

AutoAI allows you to join multiple data sets as part of the experiment set up, but since we have prepared our data already we are not going to explore that feature today. Select your newly joined data set only.

Graphical user interface, application

Description automatically generated

On the next screen, please click “No” when you are asked if you want to create a time series forecast.

Graphical user interface, application

Description automatically generated

On the next screen we will need to let AutoAI know what we would like to predict (our target). In data science terminology the columns that we use to predict churn are called features, and the column that we are trying to predict is called target.

Graphical user interface, application, Teams

Description automatically generated

AutoAI will switch to Binary Classification (CHURN is a yes/no choice – or, more precisely True/False – the customer will either leave us or will not). Let’s explore additional settings – click Explore Settings. Positive class T (True) makes sense in our case as we are looking to predict which customers will actually churn.

Graphical user interface, application

Description automatically generated

Feel free to explore the settings on the screen. However, due to time constraints, please don’t change any of the default settings. The AutoAI set up allows you to specify how many pipelines (models) you would like to build. By default, 2 top performers out of the 7 available algorithms will be used to build a total of 8 pipelines (4 for each algorithm).

Graphical user interface, text, application, email

Description automatically generated

Next, switch to the Data Source tab from the menu on the left. Explore the settings. An explanation of the most important settings can be found subsequently to the following picture.

Graphical user interface, text, application, email

Description automatically generated

When creating a model, the data is split into training and testing data. Testing data is also called “holdout data”. By default, 10% of data is used for testing. Let’s leave the default value.

*Folds* are parts of the input dataset that will alternatively be used for training and testing. By specifying 3 folds, we are creating 3 parts in the dataset. We will use the default value of folds in this lab.

When needed, we can also deselect fields to be used for training. In our case, let’s keep all of the fields.

Finally, switch to the Runtime tab and review the screen. Once done, click cancel (since you haven’t changed any of the default settings) to quit the experiment settings.

TODO: Renew picture; red mask doesn’t look good.

Graphical user interface, application

Description automatically generated

Finally, click “Run experiment” —> TODO: Let’s take a screenshot for the whole screen, and mark the button in red (to be consistent).

Icon

Description automatically generated

Watch AutoAI do its magic. This process will take approx. 2 minutes. If the experiment is taking more than 5 minutes to complete, in the interest of time, please stop the experiment manually. It will keep all the already built models but will not generate any new ones.

Graphical user interface, application, Teams

Description automatically generated

Once the process completes, Click on “swap view” to see the following charts:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, chart

Description automatically generated

A *pipeline* (in the following abbreviated with *P*) is a term used to describe various steps in creating a model. A *P* always contains the generated model. It can also contain steps to generate new columns of data to use as input variables (*features*) and steps to tune the model (*hyperparameter optimization*).

PROBLEM: Algorithms have a random starting point. Hence, The interpretation of the results varies.

As we can see in the example, *P1* and *P5*build a pipeline using the *estimators* (algorithms) which AutoAI has determined to be the best fit for the data, in this case *Snap* *Random Forest* and *XGB Classifier.*

*P2* and *P6* perform hyperparameter optimisation (HPO). Hyperparameters are “settings” (parameters) that are specific to each algorithm. Hyperparameter optimisation means that we are building the model using different settings in the algorithm. *AutoAI* tries several combinations and determines the combination which will produce the best result.

*P3*and*P7* perform feature engineering (derives new features) and build a model with these features.

*P4* and *P8* perform hyperparameter optimization for the model that uses the derived features.

In summary, for each algorithm AutoAI creates 4 pipelines:

* *P* 1 contains just the model
* *P* 2 contains the algorithm and performs HPO
* *P* 3 generates additional features and builds a model that includes these features
* *P* 4 generates the same features as pipeline 3 and performs HPO.

Once the experiment has finished or has been stopped, review the built pipelines and explore the metrics the platform generated. Switch to the Pipeline comparison tab.

Chart, line chart

Description automatically generated

Use the Rank preference button (top right on the screen) to switch between different evaluations metrics that are used by data scientists.

Graphical user interface, text, application, chat or text message

Description automatically generated

A picture containing chart

Description automatically generated

Score types: The *Holdout Score* is the score for the *test* dataset that was not used for modeling (by default, 10 percent of the data was left out for testing). The *Cross Validation Score* is the score when the cross validation technique was used. Cross validation uses different parts of the same dataset first for training, then for testing.

An experienced data scientist may look at a combination of evaluation metrics before determining if the model is ready for production, as these measures provide information on how the model performs in difference areas of “good-ness”.

**For example:**

* Precision tells you: “Out of all the records in the hold-out sample that the model predicted to be Churners (T), what percentage actually churned…”. For Pipeline 1, from all the records that the model predicted were churners, 0.896 (89.6%) were actual churners.
* Recall tells you: “Out of all the churners in the hold-out sample, what percentage can your model actually find?”. For Pipeline 1, the model managed to correctly identify 0.853 (85.3%) of the total churners in the hold-out sample.

Many data scientists use *Area Under ROC Curve.* As a rough guide, you can use the following guidelines for *Area Under ROC Curve*:

* *.90-1 = excellent (A)*
* *.80-.90 = good (B)*
* *.70-.80 = fair (C)*
* *.60-.70 = poor (D)*
* *.50-.60 = fail (F)*

Now, scroll down and review individual pipelines. In our case, AutoAI flagged Pipeline 3 as the best performing based on Accuracy (Optimized). Click on the name of the pipeline to see further details. Note: Due to randomness, your results are likely to vary.

A screenshot of a computer

Description automatically generated

Review the details. Under “Evaluation”, you will find various metrics that are familiar to data scientists and help them to evaluate which pipeline has performed in the best way. In our case, we focus on the metric “accuracy” for the purpose of ease. Depending on the business context, other metrics (like precision or recall) can be more relevant.

Graphical user interface, table

Description automatically generated

Another interesting aspect can be found under “feature summary”. Here, we can see which features make the most impact on the prediction.

Graphical user interface, application

Description automatically generated

Let’s now save the pipeline. Just click on one of the pipelines, and a “save as” button will appear – as shown below.

Graphical user interface, application

Description automatically generated with medium confidence

Please save the model as a “Model”, not as a Notebook. You do not have to change the name- this is automatically generated for you.

Graphical user interface, application, Teams

Description automatically generated

Repeat the same process. This time, please save it as a Notebook.

Graphical user interface, application, Teams

Description automatically generated

Click the “View in Project” link for the notebook you just saved.

Graphical user interface, application

Description automatically generated

Review the notebook – AutoAI is not a black box, it produces well annotated code that you can take as-is or choose to develop further.

Graphical user interface, text, application, email

Description automatically generated

Next, navigate back to the project Assets view by clicking the breadcrumb with your project name (top left of the screen). Scroll down – note that you now have your saved model and notebook listed.

A screenshot of a computer

Description automatically generated

Before moving to model deployment, let’s ensure ‘good housekeeping’ practices are followed and clean up any environments that may still be running against your project and account as they are no longer needed (this will also help minimize any potential charges and ensure you stay within the allotted CUH limits for your account type).

Please switch to the Assets tab and select Environments (See below) If there are any active runtimes still running there – please stop them at this point.

Graphical user interface, application, Teams

Description automatically generated