Article
José Pereira · Dec 21, 2021 8m read

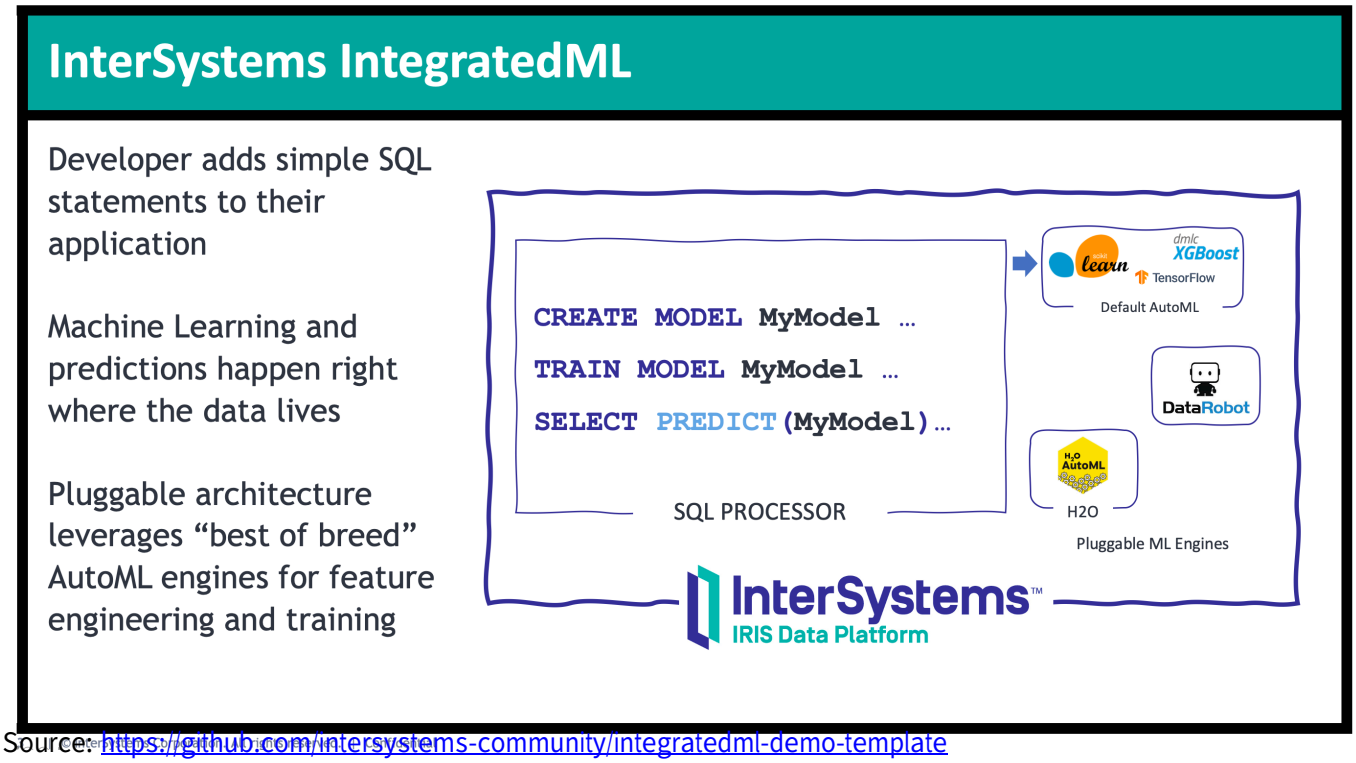
IntegratedML hands-on lab

Have you tried the InterSystems learning platform lab for IRIS IntegratedML? In that lab you can train and test a model on a readmission dataset and be able to predict when a patient will be readmitted or not, or calculate its probability of being readmitted.

You can try it without any installation on your system, all you have to do is start a virtual lab environment (Zeppelin) and play it around!

In this article we ' ll use this lab to briefly introduce you to IntegratedML, presenting you the problem to be handled, how to use IntegratedML to create a readmission prediction model, as well as some insights on how to analyze its performance metrics.

What ' s IntegratedML?



Source: https://github.com/intersystems-community/integratedml-demo-template

Before starting the tutorial, let ' s talk briefly about IRIS IntegratedML. This tool enables you to perform machine learning (ML) tasks directly in SQL statements, abstracting the implementation of complex processes like choosing which columns and ML algorithms are the best choice for performing classification or regression on a target column, for instance.

Another great functionality of IntegratedML it's easy deployment. Once your model is trained and performing well, you just need to run SQL statements in order to get your model in production.

IntegratedML gives you the choice of which ML provider to use. The default provider is AutoML, an InterSystems Python implementation using the well-known ML library scikit-learn. But you can also choose between other three providers: PMML, H2O and DataRobot.

In this article, we ' re going to use the AutoML provider.

Problem and solution

Now, let ' s introduce the problem and how we can work to propose a way to minimize its impacts. We are going to try to reduce the problems due to readmissions.

According to Wikipedia, a hospital readmission is an episode when a patient is authorized to leave the hospital (discharged), but this person returns (readmits) to the hospital again in an unexpected short time interval.

Source: https://news.yale.edu/2015/01/15/when-used-effectively-discharge-summari...

Readmissions cause losses in patients' care quality - the time between a discharge and a readmission could be critical, and hospital resources optimization as well.

One approach to overcome this problem is to try to use a historical database to create a dataset, in which past readmission episodes could be analyzed by machine learning algorithms, creating a ML model. If the dataset is rich and clean enough, readmission patterns could be correctly detected and new episodes could have their probability calculated by this model.

Thus, being able to avoid wrong discharges by using a ML model to predict readmissions, certainly will be a valuable option for hospitals to increase the quality and profit of their services.

Creating and using a ML model using IntegratedML

Model creation

Creating a ML model using IntegratedML is as easy as performing a single SQL statement. You just need to define the history dataset where the data is and a name for your model:

```
CREATE MODEL Readmission PREDICTING (MxWillReAdmit) FROM EncountersHistory
```

After that statement, you have declared and created a model called Readmission intended to predict the values for a column called MxWillReAdmit, based on a dataset called EncountersHistory.

You can find more information about this statement here. For example, in design phase, it ' s handy forces the same results on training, so you can use the USING argument with some arbitrary constant number, like this:

```
CREATE MODEL Readmission PREDICTING (MxWillReAdmit) FROM EncountersHistory USING {"se
ed": 3}
```

Model training

Now, your model is ready to be trained, by using the TRAIN MODEL statement:

```
TRAIN MODEL Readmission
```

Here, IntegratedML does a lot of work for you (using AutoML ML provider):

- Feature engineering: which columns to use
- Data encoding: how data in chosen columns must be presented to ML algorithms
- ML algorithm selection: which algorithm leads to the best results
- Model selection: based on the target column data type, a classification model or a regression model should be selected. In this example, a classification model will be selected once the target column has boolean values.

This statement could take some time, depending on the dataset size and the complexity of your data.

Model validation

By now, we should be anxious to know how good our model actually is. Integrated ML has a statement for calculating performance metrics, based on another dataset, other than the one used in training - otherwise you won ' t be fair with yourself, right? :)

This new dataset is called a testing or validation dataset, depending on which strategy you ' re using for validation. This dataset generally is retrieved from the same dataset used for creation of training dataset. A common approach is to randomly select 70% or 80% of the dataset for training and let the rest for testing/validation.

In the lab, a new ready to use dataset was previously prepared for this task: the EncountersNew dataset.

Now, we can find out how good (or bad) our model was:

```
VALIDATE MODEL Readmission FROM EncountersNew
```

In order to get the results, you should query the INFORMATIONSCHEMA.MLVALIDATIONMETRICS table:

```
SELECT * FROM INFORMATION_SCHEMA.ML_VALIDATION_METRICS
```

The model performance is measured in four metrics by IntegratedML:

- Accuracy: rate of correct predictions (values close to 1 means high correct answer rates)
- Precision: rate of correct positive predictions regarding all positive predictions did by the model (values close to 1 means fews false positives predictions)
- Recall: rate of correct positive predictions regarding all actual positives values in the dataset (values close to 1 means fews false negatives predictions)
- F-Measure: another way to measure accuracy, used when accuracy are not performing well, generally on imbalance problems (values close to 1 means high correct answer rate)

Some discussion on performance metrics

Here, we can see that the model has an accuracy of 82%. But, this metric shouldn't be analyzed alone, other metrics like precision and recall must be evaluated as well.

Let ' s think about the implications of erroneous predictions - false positives and false negatives. For our model, a false positive means that a predicted readmission wasn ' t an actual readmission. While a false negative, means that a patient had readmited and the model predicted that this patient won ' t.

Both cases lead to wrong decisions. A false positive prediction could lead to a decision of keeping a patient at the hospital more than necessary; while a false negative prediction could lead to a decision of discharge a patient early, and then readmiting that patient.

Note that, in our model, a false negative leads to readmission cases, which is what we are trying to avoid. So, we must choose models that decrease the number of false negatives, even if the number of false positives increases.

Thus, in order to choose models with low rates of false negatives, we need to choose models with high recall rates. That is because as higher the recall gets, lower the false negative is.

As our model has a recall rate of 84%, let ' s assume it as a reasonable value for now.

Making predictions

Once you have your model trained and its performance it ' s acceptable, now you can execute the model in order to predict results.

You should use the PREDICT function in order to user a model to do predictions:

```
SELECT TOP 100
    ID,
    PREDICT(Readmission) AS PredictedReadmission,
    MxWillReAdmit AS ActualReadmission
FROM
    EncountersNew
```

This function evaluates the probability of the model target column of a row being a true or a false value, based on an internal threshold.

You can see that our model made some wrong prediction - which is normal.

Another function which could be used to make predictions, is the PROBABILITY function.

```
SELECT TOP 10
    ID,
    PROBABILITY(Readmission FOR '1') AS ReadmissionProbability,
    PREDICT(Readmission) AS PredictedReadmission,
    MxWillReAdmit AS ActualReadmission
FROM
    EncountersNew
```

In the same way that PREDICTION, PROBABILITY uses the columns of a row as input for the specified model , but here we need to define which class we ' d like to calculate the probability.

This function could be useful if we ' d like to customize the threshold which is used to make the predictions.

Querying trained models

After creating and training your models, you can query for them using the INFORMATION_SCHEMA.ML_TRAINED_MODELS table, like this:

```
SELECT * FROM INFORMATION_SCHEMA.ML_TRAINED_MODELS
```

Conclusion

In this article we had created, trained, validated and made predictions on a historical patient readmission dataset. These predictions could be used as one more tool in the search for making better decisions which could improve better patient caring and reduce costs on hospital services.
All those steps were made only using the old and good SQL language through the InterSystems IntegratedML, which brings the power of machine learning to a wide branch of developers.

#IntegratedML #InterSystems IRIS