* Datasets:
  + Churn:

Predict behaviour to retain customers. You can analyse all relevant customer data and develop focused customer retention programs.

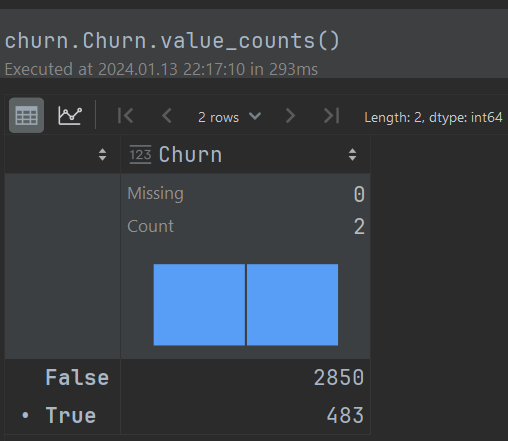
* + Digits:

The test set of the UCI ML hand-written digits datasets. The data set contains images of hand-written digits: 10 classes where each class refers to a digit.

* Imbalanced Classes

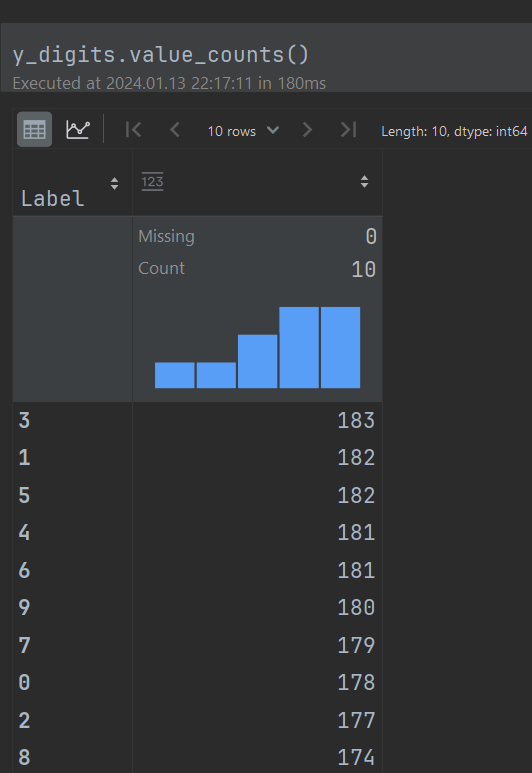
A classification data set with skewed class proportions is called imbalanced. Classes that make up a large proportion of the data set are called majority classes. Those that make up a smaller proportion are minority classes

* + Churn



Looking at the above chart we can clearly see a pattern of imbalance classes, as major and minor class.

* + Digits



Looking at the Digits dataset chart we can see that all ten classes are almost balanced.

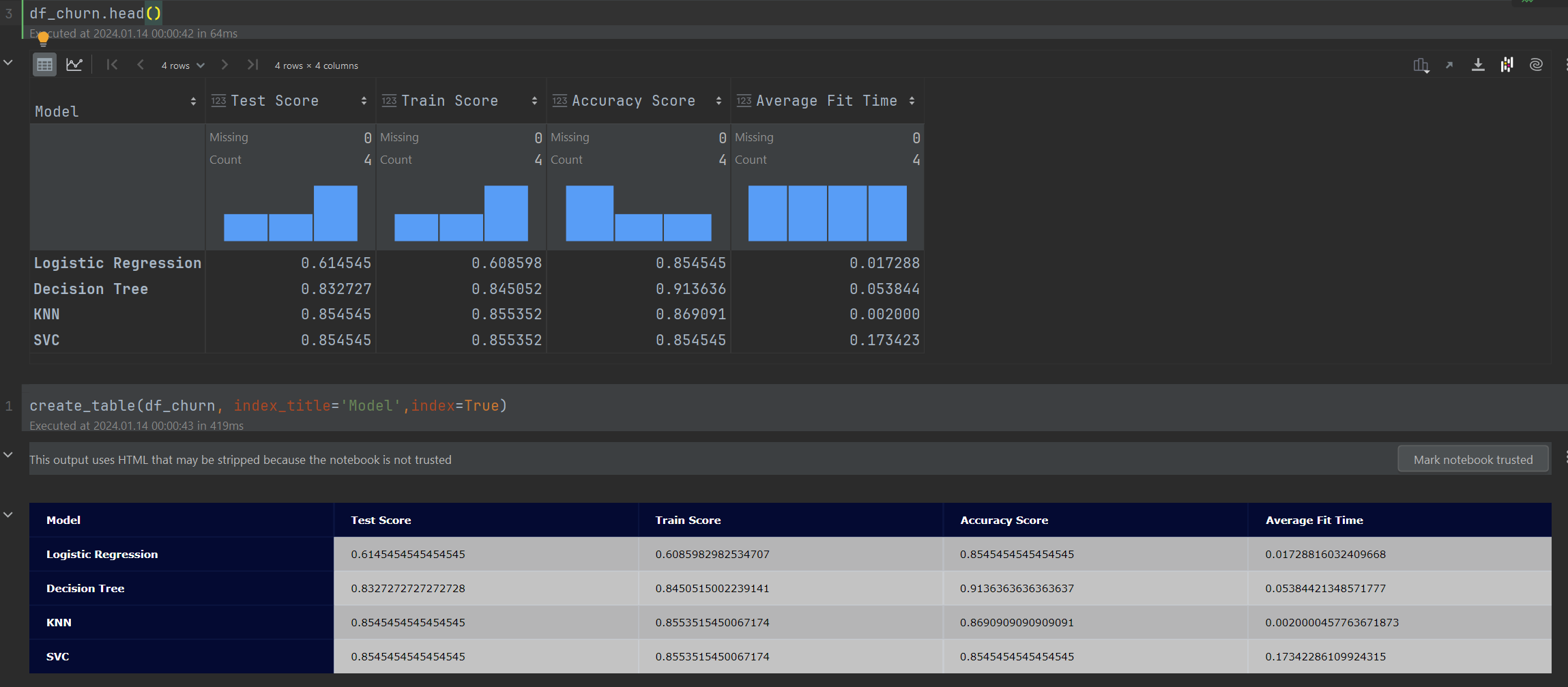
Interpretable Models

There are different ways to interpret your machine learning models. The easiest split is between interpretable models and model-agnostic methods. Interpretable models are models who explain themselves, for instance from a decision tree you can easily extract decision rules. Model-agnostic methods are methods you can use for any machine learning model, from support vector machines to neural nets. Models you can interpret are, among others, linear regression models, **logistic regression models** and **decision trees**

Model Performance

We will look at the classification model metrics and model performance by dataset.

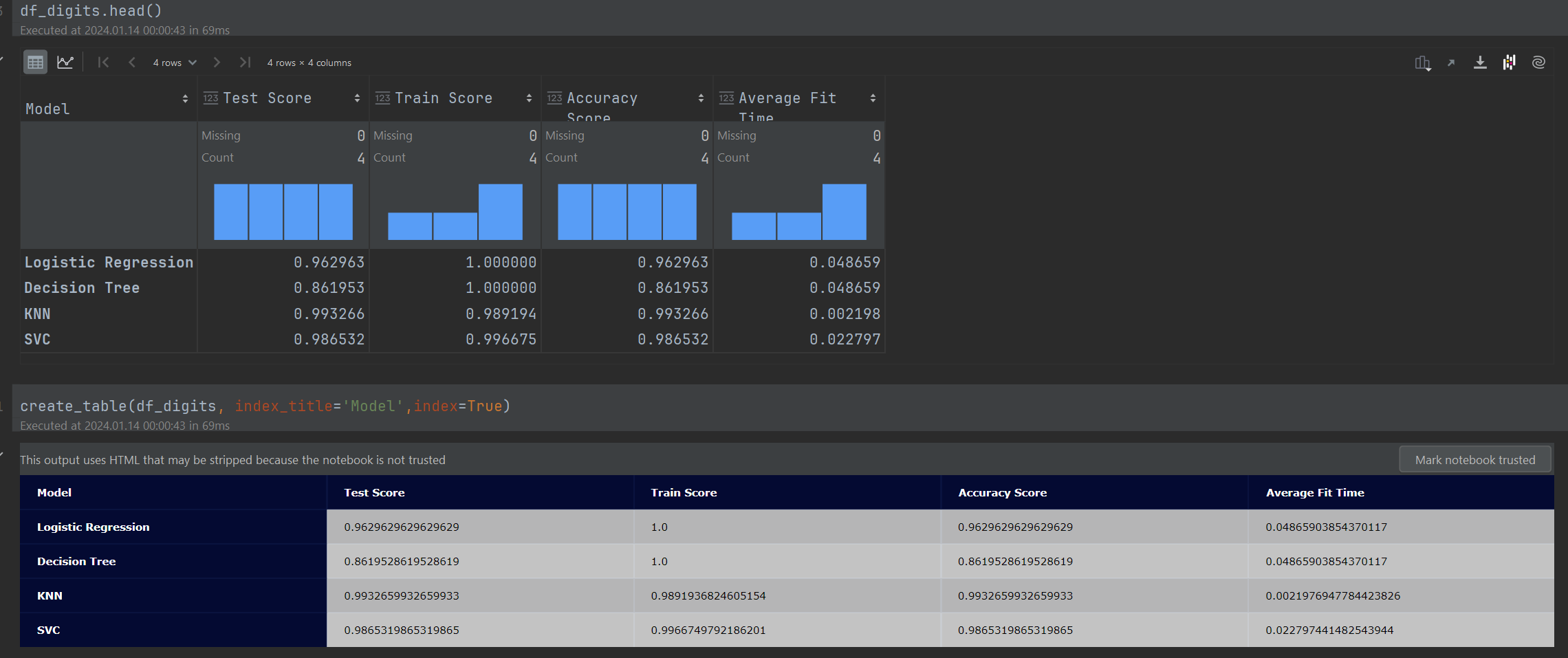
* Churn dataset



From the above Churn table, we can see that best run time measuring the average fit time with 0.002 sec is KNN model, the most accurate of the models is Decision Tree with 0.9136 scores, and the models which best explain the training and test data with 0.85 scores are KNN and SVC models.

The training and test dataset have been scaled with StandardScaler, because of imbalanced classes but the KNN and SVC models got the same score, when we test the accuracy score on the original data (unscaled) the KNN model performs better.

* Digits dataset



From the above Digits table, we observe that the fastest runtime of the average fit time is KNN model with 0.00219 sec, the model with highest accuracy score with 0.9932 is KNN model. The Logistic Regression and Decision Tree models best explain the training data with 1.0 scores and the KNN model best explain the test data with 0.9932 scores.

In conclusion, the KNeighborsClassifier (KNN) model is best performing and most accurate for the above two datasets (Churn and Digits).

According to the datasets and the chosen models, Logistic Regression and Decision Tree models might be useful for dataset with balanced classes.

SVMs can efficiently perform nonlinear classification, using a trick or parameter known as the kernel, which maps their inputs to high-dimensional feature spaces as Digits dataset which have 10 classes.

\*KNN – Nearest Neighbors Classification (KneighborsClassifier)

\*SVC - Support Vector Machines (Support Vector Classification)

\* Accuracy Score performed on whole dataset

Vincent you are right, in my case I noticed the KNN model outperforms SVC model in dataset(churn) with imbalanced classes.

Just to add, the Logistic Regression is also preferable to SVC model because of interpretability of the model.

Hey Jim, something is odd with your Logistic Regression model you have created, it does not record any difference between balanced and imbalanced dataset, also it records the longest fit time.

It is a simple model have a look at what you are passing at the constructor any hyperparameters or strange fitting data.

SVC models is mostly useful dealing with dataset having high-dimensional feature spaces, as Digits has 10 classes.

Regarding large datasets, sk-learn recommends; For large datasets consider using **[LinearSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html" \l "sklearn.svm.LinearSVC" \o "sklearn.svm.LinearSVC)** or **[SGDClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html" \l "sklearn.linear_model.SGDClassifier" \o "sklearn.linear_model.SGDClassifier)** instead, possibly after a **[Nystroem](https://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.Nystroem.html" \l "sklearn.kernel_approximation.Nystroem" \o "sklearn.kernel_approximation.Nystroem)** transformer or other [Kernel Approximation](https://scikit-learn.org/stable/modules/kernel_approximation.html#kernel-approximation).

Henry, I came to the same conclusion the Decision tree model performs very well with few classes, but it struggles with many classes, and it is very noticeable at digit dataset.

The Decision Model is the preferable model when dealing with few classes, as is the case in Churn dataset.