Azure machine learning studio tutorial pdf

What if you can predict whether your inventory by choice will increase or fall over the next month? Or if your favorite football team wins or loses the next game? How can such predictions be made? Machine learning may be part of the answer. Cortana, the new digital personal assistant powered by Bing that comes with Windows Phone 8.1, accurately predicted 15 of the 16 matches during the 2014 FIFA World Cup. In this Azure tutorial, we'll explore the features and capabilities of Azure Machine Learning by solving one of the problems we face in our daily lives. From the developer's point of view, problems can be divided into two groups - those that can be solved using standard methods, and those that cannot be solved by standard methods. Unfortunately, most of the real life problems belong to the second group. This is where machine learning comes in. The basic idea is to use machines to find meaningful patterns in historical data and use them to solve the problem. The gas price problem is probably one of the elements already in most people's budgets. A steady increase or decrease may affect the prices of other foods and services as well. There are many factors that can affect gas prices, from weather conditions to policy decisions and administrative fees, as well as to completely unpredictable factors such as natural disasters or wars. The plan for this Azure machine learning tutorial is to examine some of the available data and find correlations that can be used to create a prediction model. Azure Machine Learning Studio Azure Machine Learning Studio is an integrated web-based development environment for creating data experiments. It is closely related to other Azure cloud services and simplifies the creation and deployment of machine learning models and services. Creating an experiment There are five basic steps to creating a machine learning example. We will examine each of these steps by developing our own gas price prediction model. Obtaining data collection data is one of the most important steps in this process. The adequacy and transparency of data are the basis for creating good prediction models. Azure Machine Learning Studio provides a number of sample datasets. Another great set of datasets can be found in archive.ics.uci.edu/ml/datasets.html. After collecting the data, we need to upload it to the Studio using a simple data transfer mechanism: After uploading, we can preview the data. The image below shows some of our data that we have just submitted. Our goal here is to predict the price under the column marked e95. The next step is to create a new experiment by dragging and dropping modules from the panel on the left to the Pre-processing of data before data processing involves adapting the available data to your needs. The first module we will use here is Descriptive Statistics. Calculates statistics based on available data. Data. Descriptive Statistics module, one of the commonly used modules is Clean Missing Data. The purpose of this step is to give meaning to missing (null) values by replacing them with another value or removing them altogether. Define a function Another module used in this step in our tutorial is filter-based feature selection. This module specifies the dataset functions that are most relevant to the results we want to predict. In this case, as you can see in the photo below, the four most important features for the E95 value are EDG BS, Oil, USD/HRK and EUR/USD. Since EDG BS is another starting value that cannot be used for prediction, we will only select two of the other important features - namely the oil price and currency exchange rate in the USD/HRK column. An example of a dataset after processing is shown below: Choosing and applying the algorithm Our next step is to split the available data using split. The first part of the data (in our case 80%) will be used to train the model, and the rest is used to evaluate the trained model. The following steps are the most important steps in the entire Azure machine learning process. The Train Model module accepts two input parameters. The first is raw training data, and the second is a learning algorithm. Here we will use the Linear Regression algorithm. The train model output is one of the input parameters of the Score Model module. The second is the rest of the available data. The Result model adds a new column to our dataset, Evaluated Labels. The values in the Labels column are scored closer to the values of the corresponding E95 values when the learning algorithm used works well with the available data. The Model Evaluation module gives us an assessment of the trained model expressed in statistical values. If we look at the Determination Factor, we can conclude that there is about an 80% chance of predicting the correct price with this model. Now it is worth trying to use the neural network regression module. We will need to add new Modules Train Model and Score Model and connect the output to the existing Evaluate Model module. The Neural Regression module requires a slightly larger configuration. Since this is the most important module of the whole experiment, it is there that we should focus our efforts by tweaking and experimenting with settings and choosing the right learning algorithm as a whole. In this case, the Evaluate module gives us a comparison of our two trained models. Again, based on the determination factor, we see that neural networks provide slightly less accurate predictions. At this point, we can save selected trained models for future use. Once we have a trained model, we can continue to create a Scoring Experiment. You can do this by creating a new experiment from scratch or by using the Azure Machine Learning Studio helper. Simply select a trained model and click on Create The. The new modules that are needed here are web service input and web service output. We'll add the Project Columns module to select our input and output values. The input values are Oil and USD/HRK, and the output value is predicted in the Labels column of the Score Model output. The following figure shows our scoring tests after these esquos and when you connect the web service input modules and the web service output. Another cool helper feature comes into play at this point. You can use Web Service Publishing to create a simple web service hosted in your Azure cloud infrastructure. Anticipating new data Finally, we can test our web prediction service using a simple test form. Application This simple machine learning tutorial showed you how to create a fully functional Web prediction service. Azure Machine Learning Studio integrated with Azure can be a very powerful tool for creating data tests. In addition to Machine Learning Studio, there are other machine learning solutions, such as Orange and Tiberian. Whatever development environment you like, I encourage you to explore machine learning and find an internal data analyst. The second teaching article has been updated! The second article uses two classification algorithms to predict survival rates that interested friends can point to. Azure Machine Learning Studio is one of the best tools for ml beginners. Because you don't have to write programs at all, modules created in the studio can be called directly into machine learning using a simple drag-and-pull method, significantly reducing the learning threshold. Below, we'll briefly show you how to use Azure Machine Learning Studio in Azure and actually demonstrate an example. Manual - Environment settings Sign in to the Azure portal first, click Create resource on the left, and enter the Azure Machine Learning Studio workspace to fill in the following information. Note that when Machine Learning Studio is enabled, Azure Storage is enabled to store all data at the same time. When Azure Storage is enabled, it will be in the same resource group and will not change any keys, move locations, or delete them, which will cause the entire service to stop working. Once established, you can prompt in the upper right corner and click the prompt to enter the machine learning studio workspace. Click Run studio machine learning in additional links to go to Machine Learning Studio. If you haven't already taken off Machine Learning Studio and need to log in here, you can start using Machine Learning Studio, ingsting in blue mouse! In this example, we will use the classic Telecom material collection kaggle with very similar datasets and competitions, thanks to which we will predict whether customers will jump out of contracts (transfers) to other telecommunications companies, and this collection will be downloaded here. For this binary classification issue, Machine Learning Studio has multiple models that can be used based on data size and model characteristics. In addition to classification issues, Machine Learning Studio can also solve regression, multiple classification, grouping, and even anomaly detection. In general, there are many models available, and there is an official article teaching how to choose the right algorithm as well as a cheatstring for browsing. In this demo, we'll start an experiment in the workspace and click the plus sign in the lower-left corner to add an empty experiment. Machine Learning Studio provides drag-and-pull modules that work together to train models and predict results. By dragging the cube on the left to a flowchart similar to the workflow on the right, you can create a step-by-step machine learning model. In general, the steps to establish machine learning are as follows: columns of pure preprocessing data (data master transformation) cut the data to be used to select the data to be used to train and test the model data transfer model evaluation model evaluation model (different from the scoring model, the evaluation model calculates the equivalent of the MAE/RMSE model, providing a more intuitive method of model evaluation) above 9 points, can be done through Machine Studio, completely without having to write a program. By clicking on the plus sign below, and then clicking the dataset, you can upload the newly downloaded collection from the local side and click on the collection you just uploaded by the saved datasets > My DataSets > in order in the left menu, and drag the material and drag it to the flowchart on the right. At this point, we finished with the first step of incoming data! Next, we will make a series of choices and pre-process the data, first we will select the columns that we need to use first. There are already many variable-picking methods (Lasso/RFECV...) to cope with the feature selection, but since this article is from an ML nod point of view, we will select it manually. If possible, write another detailed introduction to Azure Machine Learning Service and azure notebook in the future. To select a variable, enter Select columns in dataset on the search page on the left The column selector to select the required fields, and the column picker also allows you to select variables based on variable data patterns, such as fields that select only numeric patterns. We use manual selection of variable fields, only some data mess and irrelevant year and day do not select, other full selection, and then click in the lower right corner of the marker to set the end of this module. Then we are going to clean up the missing values in the data, and the same module is available directly in the machine learning diversion, searching clean the missing data in the menu on the left and dragging the module to the flowchart to the right and connecting to the last module. The menu on the right can adjust the definition of missing values and how they are filled in, you can fill in the desired values directly average / median, fill in the fields and methods that we choose according to the settings, without having to change. The next step is to transform a data master that has traditionally been used in random forests for classification problems, and data patterns must be numeric values to enter into the model, so a series of data master transformations is often performed in a notebook. But there is a magic feature in Machine Learning Studio, meta edit data. This module can automatically convert selected fields to category data, no need to convert complex data types, can be fed directly into model in category data patterns, is a fairly convenient feature. Search the menu on the left to edit metadata, drag to the flowchart on the right and connect to the last module, press the Run Column Picker button on the right, and follow the figure below to select the fields to move to the category master. Then continue setting the data transformation master by selecting Set Categorical and Fields by selecting functions, and the configuration should be as follows. After you clean up your data and convert patterns, the next step is to cut the dataset into training data and test datasets. Search for split data in the left menu, drag to the flowchart on the right and connect to the last module, follow the image below in the menu on the right. We set up 70 percent of the data to belong to a training dataset, the remaining data set to a test dataset, and cut by a distorted number. If you want to make sure that the results of each slice are the same, you can set the Random seed parameter to a positive integer that is not 0. After cutting the dataset, we can enter the model we want to use! Since the last thing we are going to predict is binary classification classification, we will use the built-in fortified two-class decision tree and there are many advanced machine learning models like Light GBM and XGboost that you can import and use in Studio. Search decision tree with enhanced gain in the menu on the left and drag to the flowchart on the right. After you enter the model, you can start training the model, search for train models in the left menu, and drag them to the flowchart on the right by connecting, as shown below. Then click Run Column Picker on the menu on the right. Because the target field that we want to predict is whether the customer loses, that is, the field is deflected. Therefore, we select churn in the column picker. Pressing the tag in the lower right corner shows that the original red exclamation point has disappeared, which means that we have set ourselves up to complete the module. Next, we need to get a trained model, search for the score model in the menu on the left and connect, as shown below. This way of combining means that we train a model that is trained to predict data using data from a test dataset, and there are two lines linked to the result model. When the result is complete, the final step is to evaluate the model. Search for the Review model in the left menu, drag right and connect as shown below and you're done with all the settings! Above, after nine configuration modules, we finally finished all the pre-job, and finally press a few buttons, you can let the whole service run! Press the Save key in the black menu below to archive, and then start, and the entire service will start performing operations and the screen will become the image below. After the operation returns to the original image, in the upper right corner of the flowchart will be a green marker, each module also has a green marker, representing all modules are ready. If you plan to change or attach to any of the modules, be sure to archive it again and restart it. Then press the right button on the last evaluate model module and select the evaluation results > Visualize sequently to see the final predictions for the model. The figure above, calculated using Studio, contains a beautiful ROC curve, a matrix of mistakes, the accuracy of /Precision/Recall, and the final F1 result. You can even see the distribution of predictive data patterns. What if you want to look at the forecasts for each data? In the previous Score model module, when you right-click and select the evaluation results > visualize sequently, you can see predictions for each data. In the conclusions above, we demonstrated a collection of telecommunications customer cancellation rates using Azure Machine Learning Studio, which predicts whether customers will lose through an advanced machine learning studio service. But is that the only thing Studio does? In the next article, we'll teach you how to deploy a trained model in a web app so everyone can use the model! If you think this series of articles will help you, do not hesitate to clap your hands or leave a message, your support my biggest motivation for writing Medium. maximum power. Power.