Model Explainers in PySpark

What is a model explainer?

SHAP

SHAP is one of the best model explainers for Python (and really, most ML is done in Python).

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. (Shapley values are a solution concept in cooperative game theory).

SHAP Values are used in the SHAP model explainer to show how each feature and value correlate the model output.

SHAP has the ability to show the same model summaries that the built in feature importance functions can show. However, the real power of model explainers comes in with the ability to show each individual observation.

We'll discuss this more as we get into some sample code.

DALEX

Besides SHAP, there are more model explainer packages. Dalex is one of the popular package that you can use to explain your machine learning model. You can use Dalex to do some similar things as SHAP, but you can create more insightful graphs with Dalex.

We'll discuss this more as we present some sample code.

What do we do with the info a model explainer gives us?

Once we have a model what information can we use it for. One of the most important parts of a model explainer is how we are able to see how individual features effect the outcome of the model. In the example of the being approved for a loan we could see if a feature like 'years of credit history' could negativly affect if someone is going to be approved for a loan.

In the example of our model we can look at the shap values for the amount of buisnesses in a track and see how when we have more buisnesses in a track it has a strong positive atribute to the number of restrants we have in a track. Using model explainers like SHAP allows us to see if we have some information leakage from our target varialbe.

Let's look at the example code in Databricks

The link can be here.
Alternatively, you can find the link to the code in our GitHub template or in Databrick>Worspace>Shared>Team Presentations>Model Explainers

Fin