Matt Downing and Gene Eagle – Project 2 Write Up

WHAT IS EXPLAINABLE AI?

Explainable AI (XAI) is a field of AI that focuses on model explainability. Machine Learning models are mostly black boxes - data goes in, some function explaining that data is learned, and predictions are output. We know what these predictions are and even how accurate they are, but we do not know why the model generated those predictions. As model complexity increases model interpretability decreases. This is readily apparent in Deep Learning where models can consist of hundreds of layers and contain millions of features. XAI aims to create a set of tools that produce explainable models as well as easy to understand interfaces that enables humans to interrogate, interpret and ultimately trust model output.

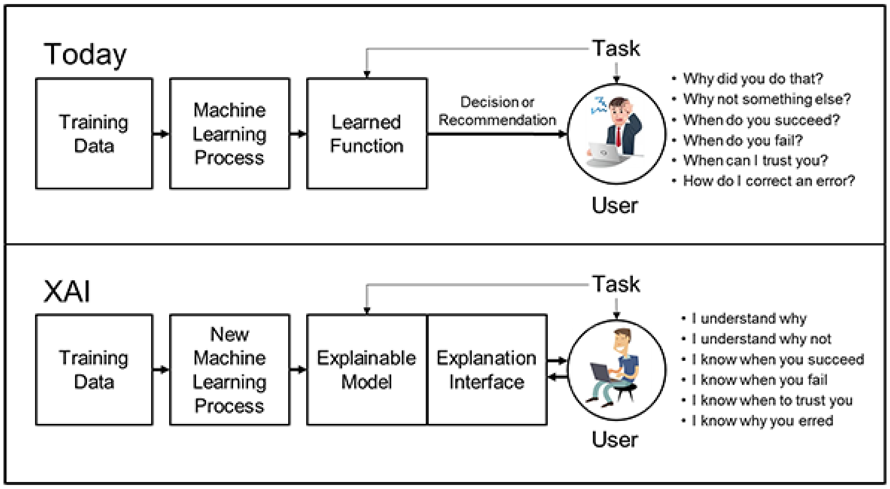

Figure 1: XAI Concept (from DARPA website)

WHY IS IT IMPORTANT?

As machine learning and artificial intelligence algorithms continue to become more complex being able to explain why the model predicted certain outcomes becomes more important. Because neural networks are highly complex it is often impossible for humans to determine why the model is outputting the predictions that it is.

Model explainability is important because it promotes trust, transparency, and accountability in the model and in the ways in which the model was built. This is especially important in mission critical domains where humans are taking an action based on a models prediction. Disease diagnosis, criminal justice, risk management and many other industries currently utilize AI to make decisions that have real life altering consequences for individuals and society as a whole.

HOW IT IS USED?

XAI provides users with an explanatory model that lives alongside the blackbox AI model. This explanatory model provides the user an explanation of what is happening inside the model - an explanation of which features are most relevant to the model. The human user can then investigate and determine if the model is performing correctly, what features are most important to the model and if those features should be used in the model.

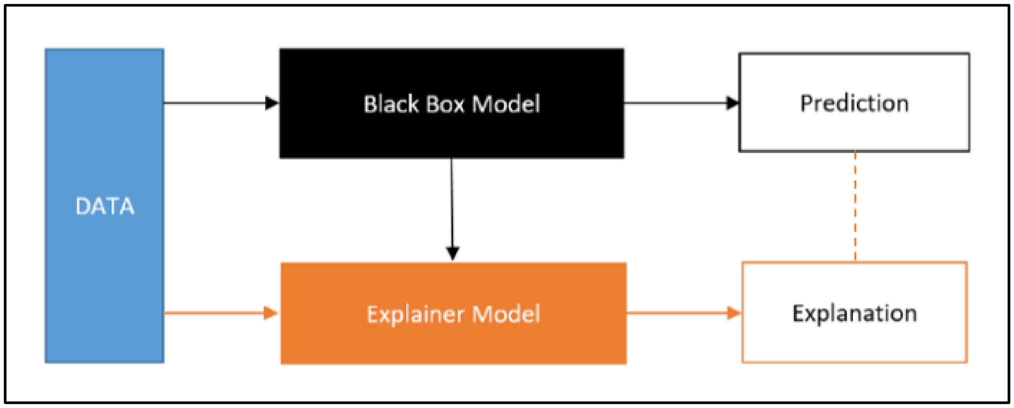

Figure 2: PyCon Sweden :Interpreting ML Models Using SHAP by Ravi Singh

SHAPELY VALUES AND SHAP APPROACH

In order to understand SHAP some brief background on Shapely Values is needed. Shapely Values come from game theory and tell us how to fairly distribute the payout among the players in a game based on how much each player contributed to the total payout in a cooperative style game. There are four rules to follow to achieve a fair payout:

- Efficiency – The sum of the Shapley values of all players equals the value of the total coalition
- Symmetry – All players have a fair chance to join the game
- Dummy – If a player contributes nothing to a coalition, then the contribution of that player is 0
- Additivity – For two outcome values, the Shapley values of the sum of the two outcome functions should be the sum of the Shapley values of the two outcome functions. This allows for simple summation. [https://storage.googleapis.com/cloud-ai-whitepapers/AI%20Explainability%20Whitepaper.pdf]

In an AI context this “game” is the prediction task for a single instance in the dataset, the “payout” is the actual prediction minus the average prediction for all instances and the “players” are the features in the dataset. The Shapley value for each feature of each individual prediction helps us interpret how much each feature contributed to the prediction - both positive and negative contribution.

Each feature (petal length, sepal length, color if using the Iris dataset) have formed a coalition to work together in order to produce a prediction. In general a Shapely value for a particular feature is calculated by simulating a possible coalition, predicting the target value, replacing that particular feature with a different value from the dataset, predicting this new coalitions target value, and finally take the difference between these predictions – this represents the marginal contribution of this particular feature. It is important to note that when creating these feature coalitions sometimes each feature is present and sometimes each feature is absent. If we do this for all possible coalitions, we can then take the average of all marginal contributions for all possible coalitions which gives us the Shapley Value for that particular feature.

This method is very computationally expensive because the computation time increases exponentially with the number of features. In real world scenarios the Shapely value is estimated by taking a sample of possible coalitions and/or by limiting the number of iterations.

What IS SHAP

SHAP (Shapley Additive Explanations) is an estimation approach based on Shapely values and can be used to explain the output of any model. SHAP quantifies the contribution each feature makes to the prediction on an individual observation basis. SHAP has the ability to give the user both the local and global interpretability:

- Global Interpretability – the collective SHAP values can be used to show how much each predictor contributes to the prediction similar to a variable importance plot. This can show us the positive or negative relationship the feature had to the prediction.
- Local Interpretability – each prediction gets its own SHAP value. This is useful because it lets the user see on a prediction by prediction basis what the model used to make that prediction.

Because true Shapley values would be infeasible to compute SHAP instead estimates approximate Shapley values for each feature using the KernalSHAP approach described below:

- Sample coalitions
- Get the prediction for each coalition
- Compute the weight for each coalition
- Fit the weighted linear regression model
- Return the Shapley values and the coefficients from the linear regression model

SHAP FOR DEEP LEARNING

For image classification SHAP creates super pixels. Super pixels are created by combining pixels of similar color and brightness into larger homogenous regions. Each super pixel represents a single feature in the model. These super pixels are then turned on and off (1,0) to form various coalitions as in the Figure 3:

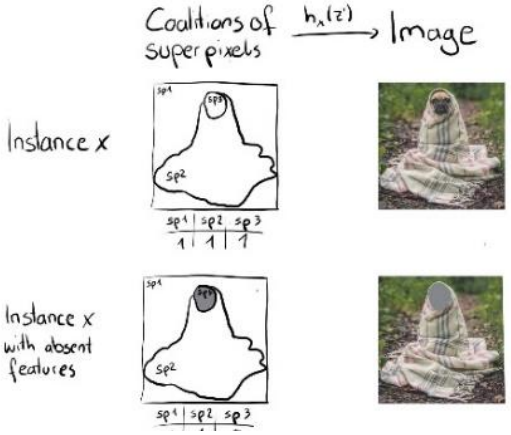

Figure 3 https://christophm.github.io/

Using Figure 3 as an example three super pixel regions were formed. For a more concrete example we can use the SHAP output for the MNIST dataset:

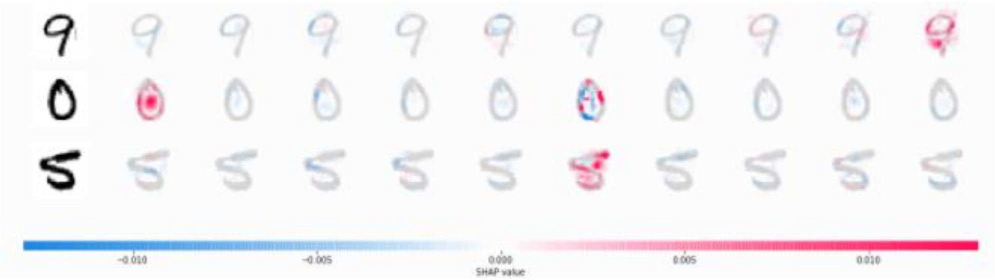

Figure 4: SHAP output for MNIST

The red regions for each observation represent the pixels that contributed positively to the classification and the blue regions contributed negatively. Note that in some cases these red regions cover areas that do not cover any pixels that contain the MNIST number. This makes sense because an absence of data is equally important as having data and can lead to a classification. This is very apparent in the '0' column where the center of the 0 is very important to the model prediction. The reverse is also true of the blue regions. Typically these blue regions fall in areas where each number is present. These areas would not help a model correctly predict a digit because most digits have this region present/filled in.