# Independent Study IMT 600

## Machine Learning Expailnability

Machine learning models are often referred to as "black boxes", in the sense that they can make good predictions, but you can't understand the logic behind those predictions.

Linear regression is a machine learning model which is commonly used as it's easier to interpret. Let's take a look at an example where we predict the price of a car using its various features like engine size, horsepower, length of car etc.

$$price = -85090 + 102.85 * engineSize + 43.79 * horse\ power + 1.52 * peak\ RPM - 37.91 * length + 908.12 * width + 364.33 * height$$

In the above model, we can see that the coefficient for width is 908.12. It means that if the width is increased by one unit, the average price of car **increases** by $908.12, with all other features held constant. Hence width contributes most towards predicting the average price of the car as it has the largest absolute coefficient of all the features. Similarly, height of the car and engine size are the next two features which are important.

As you build more complex models, it becomes much harder to interpret the models. It becomes difficult to answer questions like:

- What features in the data did the model think are most important?
- For any single prediction from a model, how did each feature in the data affect that particular prediction?
- How does each feature affect the model's predictions in a big-picture sense (what is its typical effect when considered over a large number of possible predictions)?

## Why are these questions/insights valuable?

1. *Debugging*: Understanding the patterns a model is finding will help you identify when those are at odds with your knowledge of the real world, and this is typically the first step in tracking down bugs.
2. *Feature Engineering*: It is usually the most effective way to improve model accuracy. If you have 100s of features, knowing the most important features can help you engineer new effective features.
3. *Directing Future Data Collection*: Data collection can be expensive. Model-based insights give you a good understanding of the value of features which will be helpful in future data collection.
4. *Informing Human-Decision Making*: In some cases, inference is more important than predictions.
5. *Building Trust*: To deploy a model to production, you need to trust the model to make accurate prediction. Trust comes from understanding how the model works.

## Machine Learning Explainability Techniques

I will be covering 3 techniques that will be used to explain ML models.

## Permutation Importance

One of the most basic questions we might ask of a model is: What features have the biggest impact on predictions?

*Permutation Importance* is used to calculate the feature importance of a model after it has been fit to the data. It ranks the features from the most important to the least important.

**How it works**

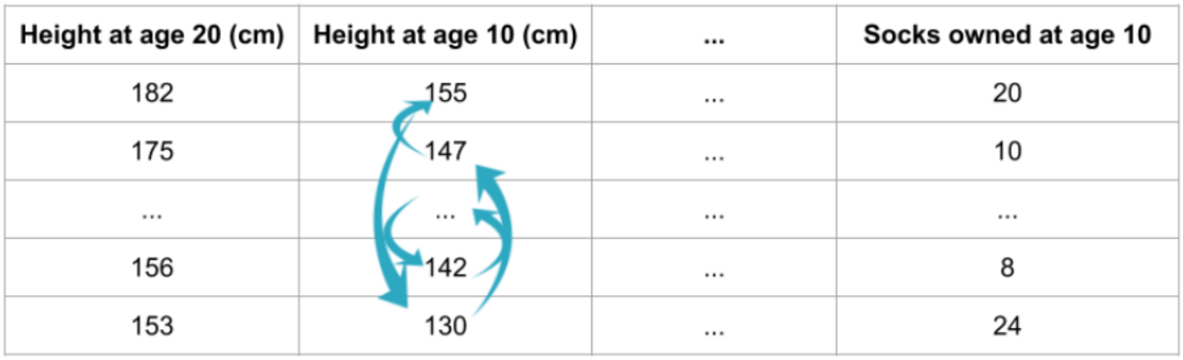
Permutation importance is calculated after a model has been fitted. A single column/feature of the validation data is shuffled randomly, leaving the target and all other columns/features in place and we then assess its affect on the accuracy of predictions on the new shuffled data.

Consider data with the following format:

| Height at age 20 (cm) | Height at age 10 (cm) | ... | Socks owned at age 10 |
|---|---|---|---|
| 182 | 155 | ... | 20 |
| 175 | 147 | ... | 10 |
| ... | ... | ... | ... |
| 156 | 142 | ... | 8 |
| 153 | 130 | ... | 24 |

We want to predict a person's height when they become 20 years old, using data that is available at age 10. We fit a linear model to predict Height at age 20 and measure the accuracy of the model.

We then implement permutation importance which shuffles data in a column and then measures the accuracy of the model keeping everything else the same. This process is repeated for all the other columns in the dataset.



| Height at age 20 (cm) | Height at age 10 (cm) | ... | Socks owned at age 10 |
|---|---|---|---|
| 182 | 155 | ... | 20 |
| 175 | 147 | ... | 10 |
| ... | ... | ... | ... |
| 156 | 142 | ... | 8 |
| 153 | 130 | ... | 24 |

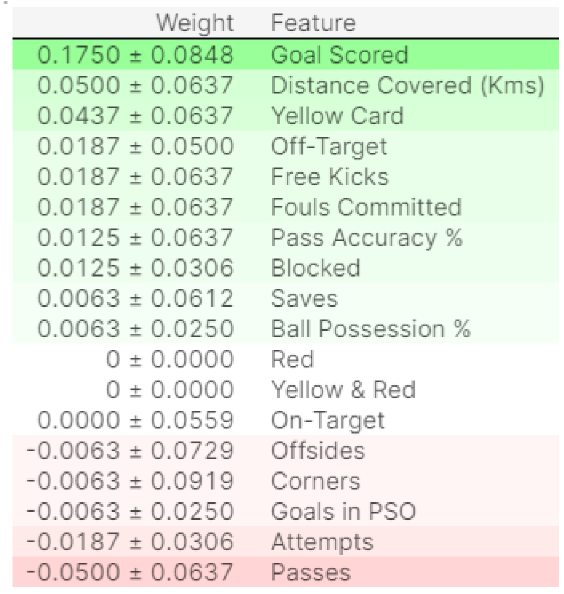
Randomly re-ordering a single column should cause less accurate predictions, since the resulting data no longer corresponds to anything observed in the real world. Model accuracy especially suffers if we shuffle a column that the model relied on heavily for predictions. In this case, shuffling height at age 10 would cause terrible predictions. If we shuffled socks owned instead, the resulting predictions wouldn't suffer nearly as much.

With this insight, the process is as follows:

1. Get a trained model.

2. Shuffle the values in a single column, make predictions using the resulting dataset. Use these predictions and the true target values to calculate how much the loss function suffered from shuffling. That performance deterioration measures the importance of the variable you just shuffled.
3. Return the data to the original order (undoing the shuffle from step 2). Now repeat step 2 with the next column in the dataset, until you have calculated the importance of each column.

Permutation importance can be implemented using the Python library eli5. I implemented this technique on a model which was built on the 2018 FIFA dataset which predicts whether a player from a team won the Man of the Match award. The output would be as follows:



| Weight | Feature |
|---|---|
| 0.1750 ± 0.0848 | Goal Scored |
| 0.0500 ± 0.0637 | Distance Covered (Kms) |
| 0.0437 ± 0.0637 | Yellow Card |
| 0.0187 ± 0.0500 | Off-Target |
| 0.0187 ± 0.0637 | Free Kicks |
| 0.0187 ± 0.0637 | Fouls Committed |
| 0.0125 ± 0.0637 | Pass Accuracy % |
| 0.0125 ± 0.0306 | Blocked |
| 0.0063 ± 0.0612 | Saves |
| 0.0063 ± 0.0250 | Ball Possession % |
| 0 ± 0.0000 | Red |
| 0 ± 0.0000 | Yellow & Red |
| 0.0000 ± 0.0559 | On-Target |
| -0.0063 ± 0.0729 | Offsides |
| -0.0063 ± 0.0919 | Corners |
| -0.0063 ± 0.0250 | Goals in PSO |
| -0.0187 ± 0.0306 | Attempts |
| -0.0500 ± 0.0637 | Passes |

**Interpreting Permutation Importance**

The values towards the top are the most important features, and those towards the bottom matter least.

The first number in each row shows how much model performance decreased with a random shuffling (in this case, using "accuracy" as the performance metric).

There is some randomness to the exact performance change from a shuffling a column. We measure the amount of randomness in our permutation importance calculation by repeating the process with multiple shuffles. The number after the **±** measures how performance varied from one-reshuffling to the next.

You'll occasionally see negative values for permutation importances. In those cases, the predictions on the shuffled (or noisy) data happened to be more accurate than the real data. This happens when the feature didn't matter (should have had an importance close to 0), but random chance caused the predictions on shuffled data to be more accurate. This is more common with small datasets, like the one in this example, because there is more room for luck/chance.

In our example, the most important feature was **Goals scored**. That seems sensible.

## Partial Dependence Plot

While feature importance shows what variables most affect predictions, partial dependence plots show ***how*** a feature affects predictions.

This is useful to answer questions like: Controlling all other features, what impact does different goals scored have on Man of the Match being awarded?

If you are familiar with linear or logistic regression models, partial dependence plots can be interpreted similarly to the coefficients in those models. Though, partial dependence plots on sophisticated models can capture more complex patterns than coefficients from simple models.

**How it works**

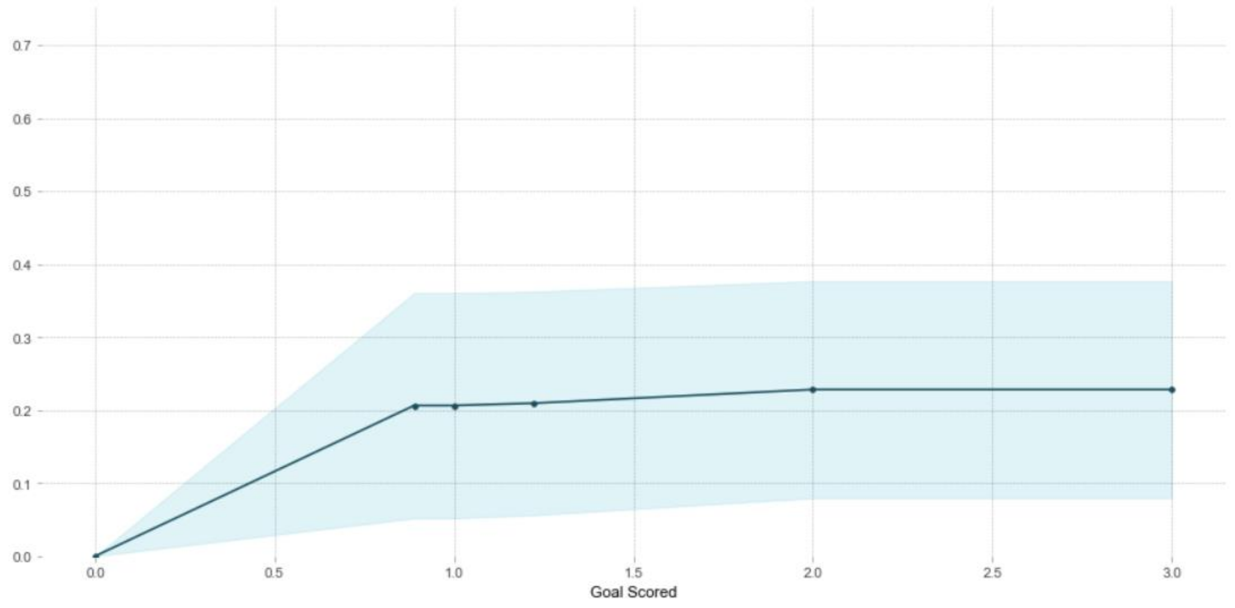
In our soccer example, teams may differ in many ways. How many passes they made, shots they took, goals they scored, etc. At first glance, it seems difficult to disentangle the effect of these features.

To see how partial plots separate out the effect of each feature, we start by considering a single row of data. For example, that row of data might represent a team that had the ball 50% of the time, made 100 passes, took 10 shots and scored 1 goal.

We will use the fitted model to predict our outcome (probability their player won "man of the match"). But we **repeatedly alter the value for one variable** to make a series of predictions. We could predict the outcome if the team scored only 1 goal. We then predict with them having scored 2 goals. Then predict again for 3 goals. And so on. We trace out predicted outcomes (on the vertical axis) as we move from small values of goals scored to large values (on the horizontal axis).

In this description, we used only a single row of data. Interactions between features may cause the plot for a single row to be atypical. So, we repeat that experiment with multiple rows from the original dataset, and we plot the **average predicted** outcome on the vertical axis. This algorithm can be implemented using the pdpbox library.

Partial Dependence Plot for Goal Scored:

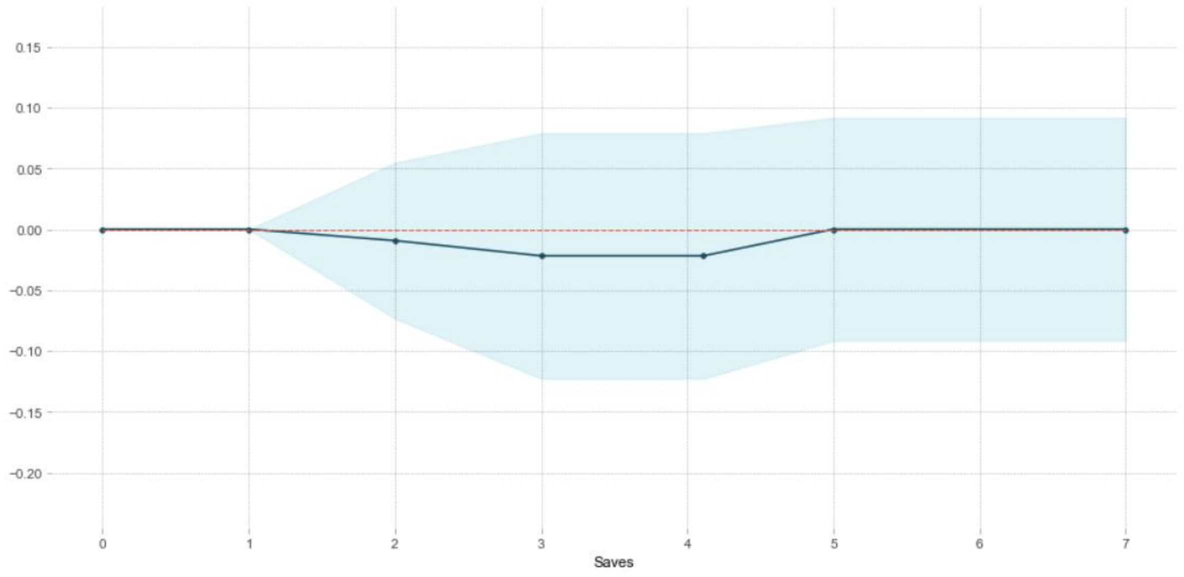


Interpreting this plot:

- The y axis is interpreted as **change in the prediction** from what it would be predicted at the baseline.
- A blue shaded area indicates level of confidence

From this particular graph, we see that scoring a goal substantially increases your chances of winning "Man of The Match." But extra goals beyond that appear to have little impact on predictions.

Partial Dependence Plot for Saves:



As you can see, Saves made does not contribute much to your chances of winning 'Man of The Match'. This is also corroborated by the fact that permutation importance technique had ranked it as not an important feature in making the prediction.

## SHAP Values

SHAP Values (an acronym from SHapley Additive exPlanations) break down a prediction to show the impact of each feature. Where could you use this?

- A model says a bank shouldn't loan someone money, and the bank is legally required to explain the basis for each loan rejection
- A healthcare provider wants to identify what factors are driving each patient's risk of some disease so they can directly address those risk factors with targeted health interventions

### How it works

SHAP values interpret the impact of having a certain value for a given feature in comparison to the prediction we'd make if that feature took some baseline value.

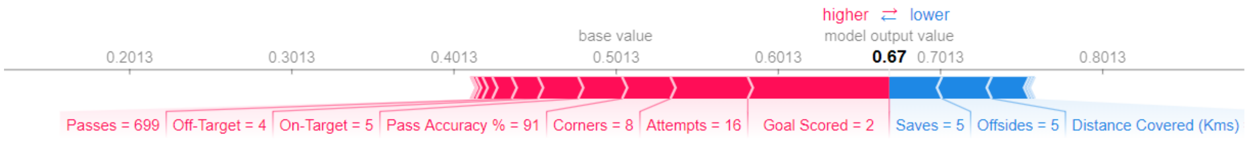
This is useful to answer questions like: How much was a prediction driven by the fact that the team scored 3 goals, **instead of some baseline number of goals.**

Of course, each team has many features. So, if we answer this question for number of goals, we could repeat the process for all other features.

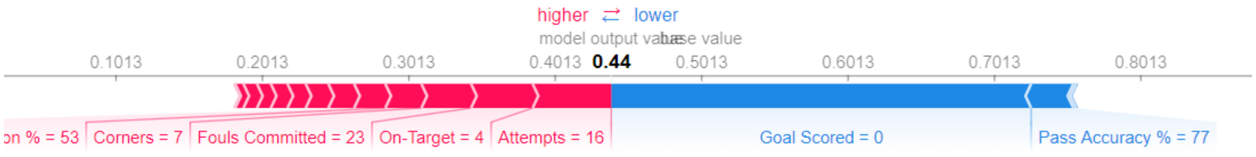
SHAP values do this in a way that guarantees a nice property. Specifically, you decompose a prediction with the following equation:

sum(SHAP values for all features) = pred_for_team - pred_for_baseline_values

That is, the SHAP values of all features sum up to explain why my prediction was different from the baseline. This allows us to decompose a prediction in a graph like this for the 3rd observation in validation set:



Similarly, the SHAP values for the prediction of the 7th observation in the validation set:



Interpreting this plot:

We predicted 0.67 for the 1st plot, whereas the base_value is 0.5013(common for all observations). Feature values **causing increased predictions are in red**, and their **visual size shows the magnitude of the feature's effect**. **Feature values decreasing the prediction are in blue**. The biggest impact comes from Goal Scored being 2.

If you subtract the length of the blue bars from the length of the pink bars, it equals the distance from the base value to the output.

For the second plot, you can see that Goal Scored feature decreases the prediction since it's 0.

## Advanced use of SHAP Values

We started by learning about permutation importance and partial dependence plots for an overview of what the model has learned.

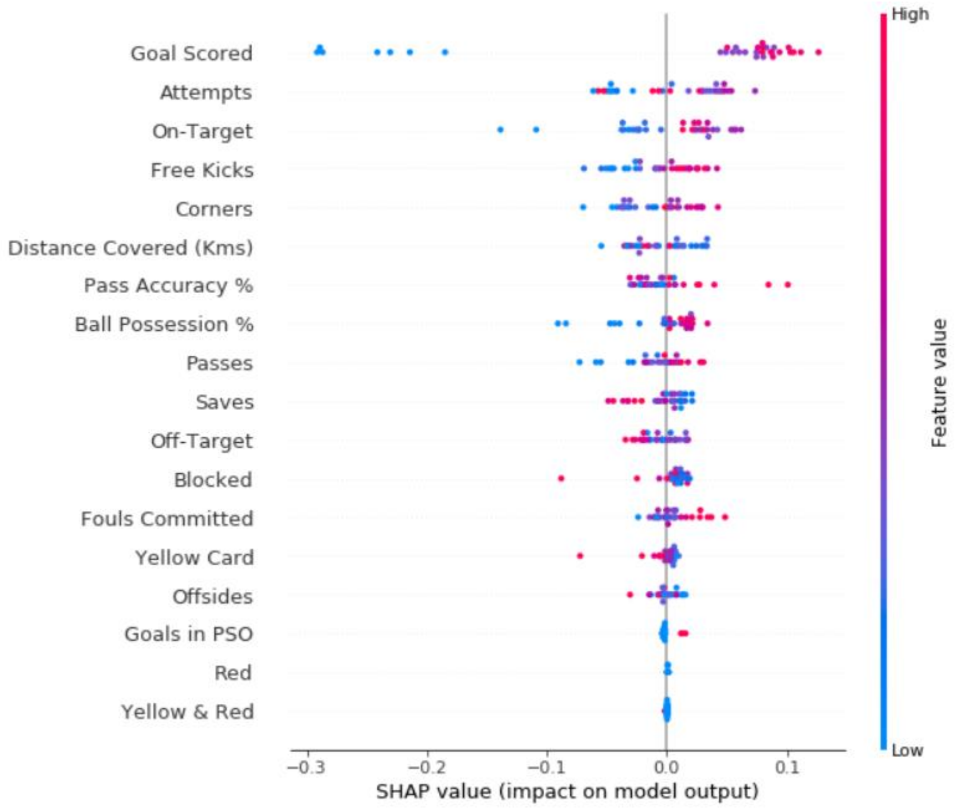
We then learned about SHAP values to break down the components of individual predictions.

Now we'll expand on SHAP values, seeing how aggregating many SHAP values can give more detailed alternatives to permutation importance and partial dependence plots.

**Summary Plots**

Permutation importance is great because it created simple numeric measures to see which features mattered to a model. This helped us make comparisons between features easily, and you can present the resulting graphs to non-technical audiences. But it doesn't tell you how each features matter.

SHAP summary plots give us a birds-eye view of feature importance and what is driving it.



Interpreting the plot:

This plot is made of many dots. Each dot has three characteristics:

- Vertical location shows what feature it is depicting
- Color shows whether that feature was high or low for that row of the dataset
- Horizontal location shows whether the effect of that value caused a higher or lower prediction.

For example, the point in the upper left was for a team that scored few goals, reducing the prediction by 0.25.