Ensemble Learning

Sebastien Perez

Agenda

Ensemble Learning:

- Voting/Averaging or Stacking
- Bagging
- Random Forest
- Gradient Boost

And also:

- Feature Preprocessing
- Machine Learning Pipeline
- Model Interpretability
- AutoML

What is ensemble learning ?

We call Ensemble Learning to the use of combination of simple models to create new models with better results than the simple ones.
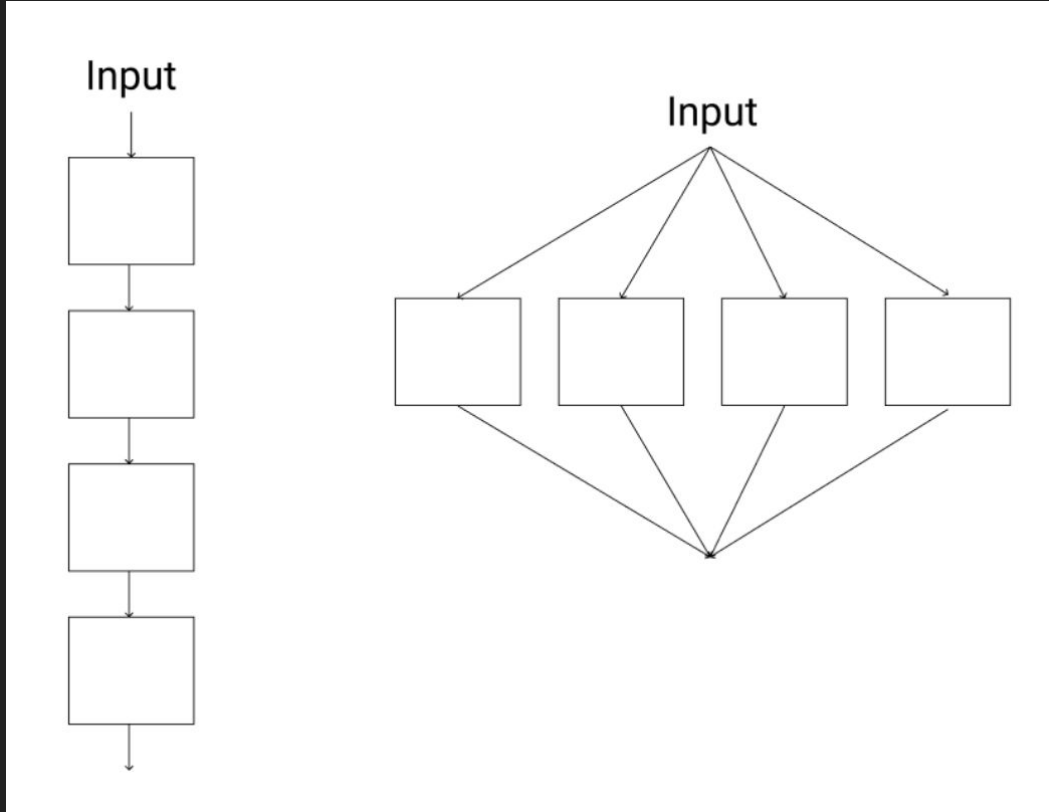
They can be used for Regression or Classification

Why is it useful to combine ?

We can see that different models

make different predictions

that can differ (we see here

output correlation)
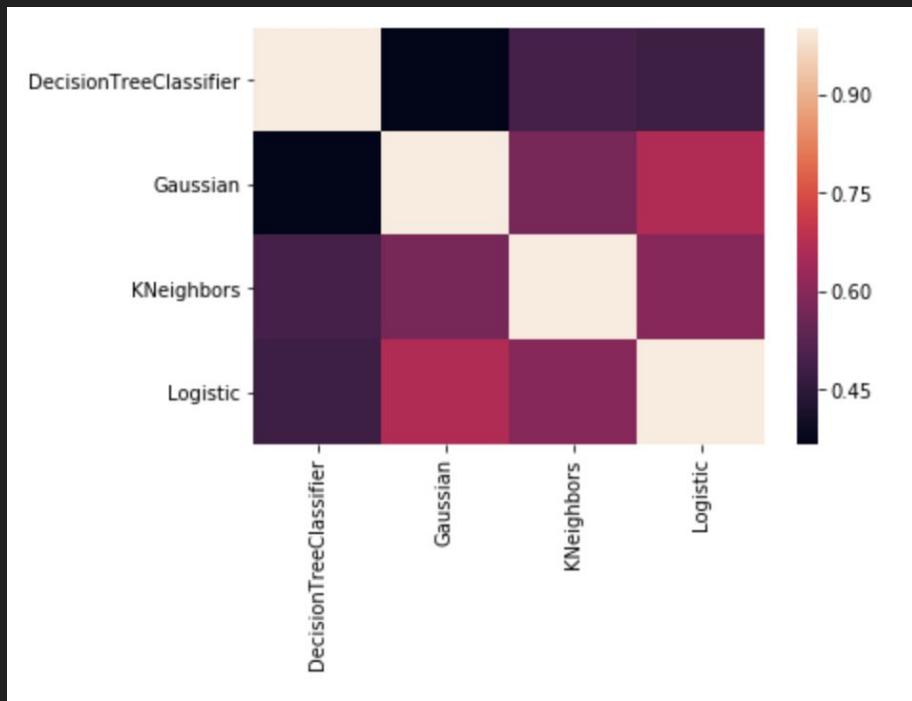
Types of Combination: Sequential vs Parallel

Types of Combination: Sequential vs Parallel

Parallel Combination: We gather a group of experts in a room and ask them at the same time their opinion

Sequential Combination: We take 1 expert, ask for her prediction and calculate her mistake. We ask a new expert to predict this time the correction needed. Then the next one is asked the correction from the 2 previous experts ...

Voting / Averaging

Models are combined by taking the vote of each one or averaging the output of each model.

Empirically, it does not give best results.

Stacking

Instead of simply averaging or voting, we can input the models prediction in a new machine learning model.

Bagging
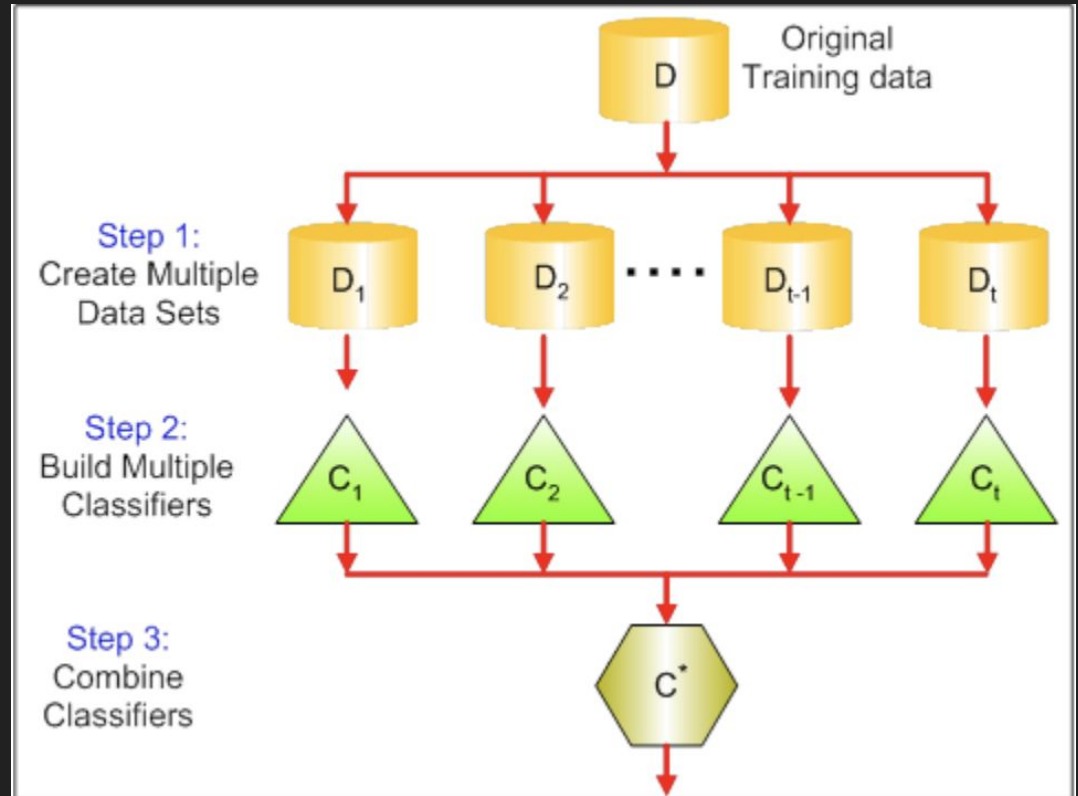
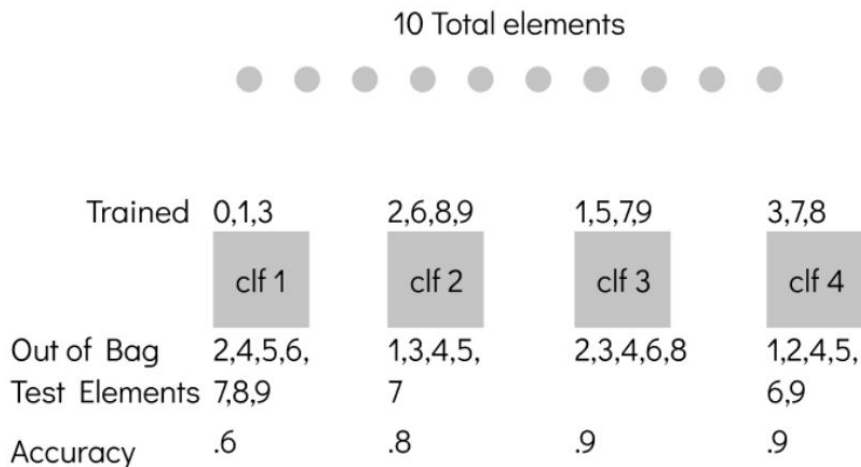Bagging consists in building **several random datasets** from the original one (different lines and features in each one) and using each one to train **simple models** (combined with average or vote).

Step 1:
Create Multiple Data Sets

Step 2:
Build Multiple Classifiers

Step 3:
Combine Classifiers

Original Training data

Out of Bag Error

Random Forest

It's bagging with Decision Trees !

Random Forest in Sklearn

```python
# Load the library

from sklearn.ensemble import RandomForestClassifier

# Create an instance

clf = RandomForestClassifier(max_depth=4)

# Fit the data

clf.fit(X,y)
```
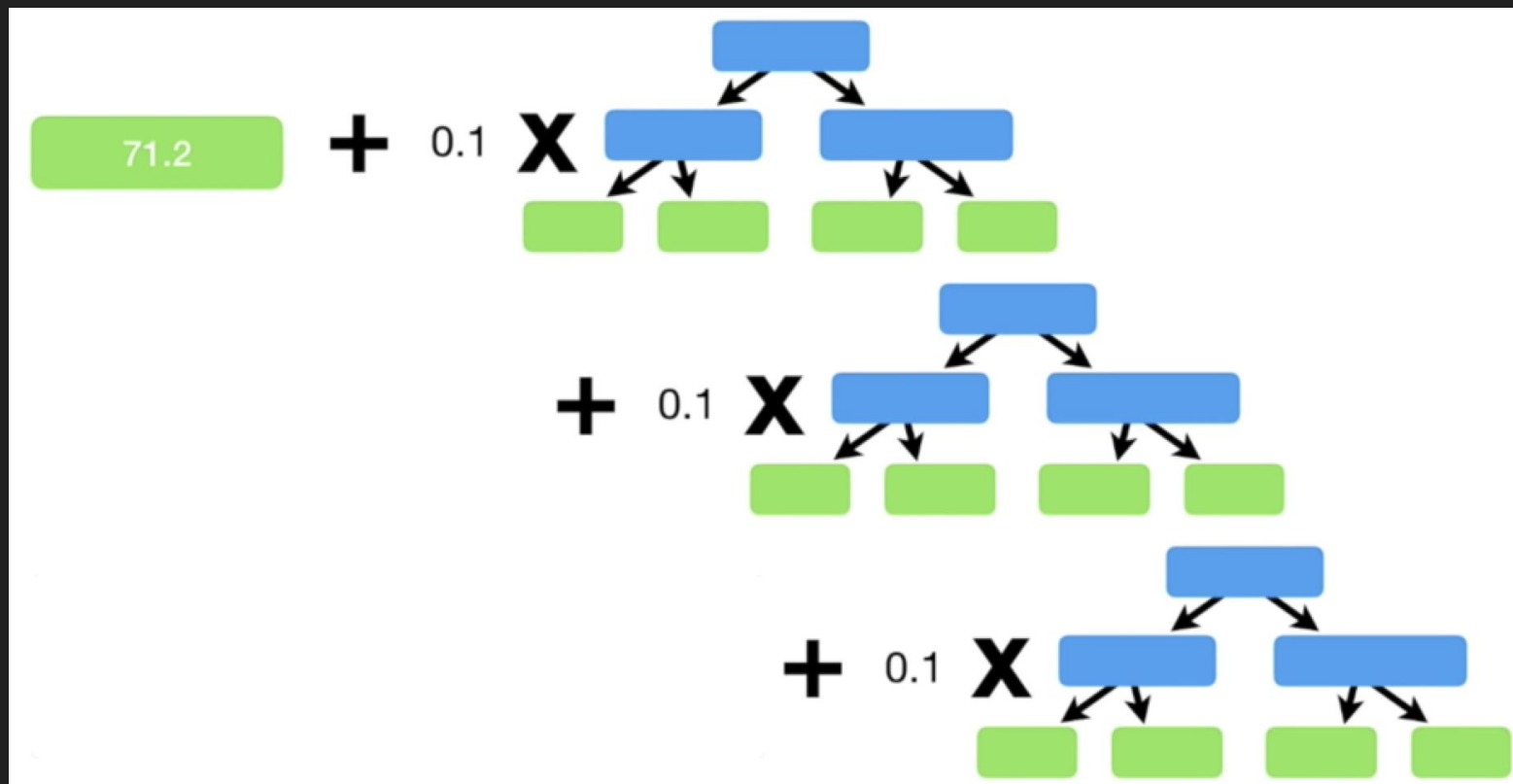
Random Forest Main Parameters

N_estimators: Number of trees

Max_depth: Number of Splits

Min_samples_leaf: Minimum number of observations per leaf

Gradient Boosted Trees

Gradient Boosted in Sklearn

N_estimators: Number of trees

learning_rate: Learning Rate of the Boosted Tree

Max_depth: Number of Splits

Min_samples_leaf: Minimum number of observations per leaf

Gradient Boosted Parameters

```python
# Load the library

from sklearn.ensemble import GradientBoostingClassifier

# Create an instance

clf = GradientBoostingClassifier(max_depth=4)

# Fit the data

clf.fit(X,y)
```
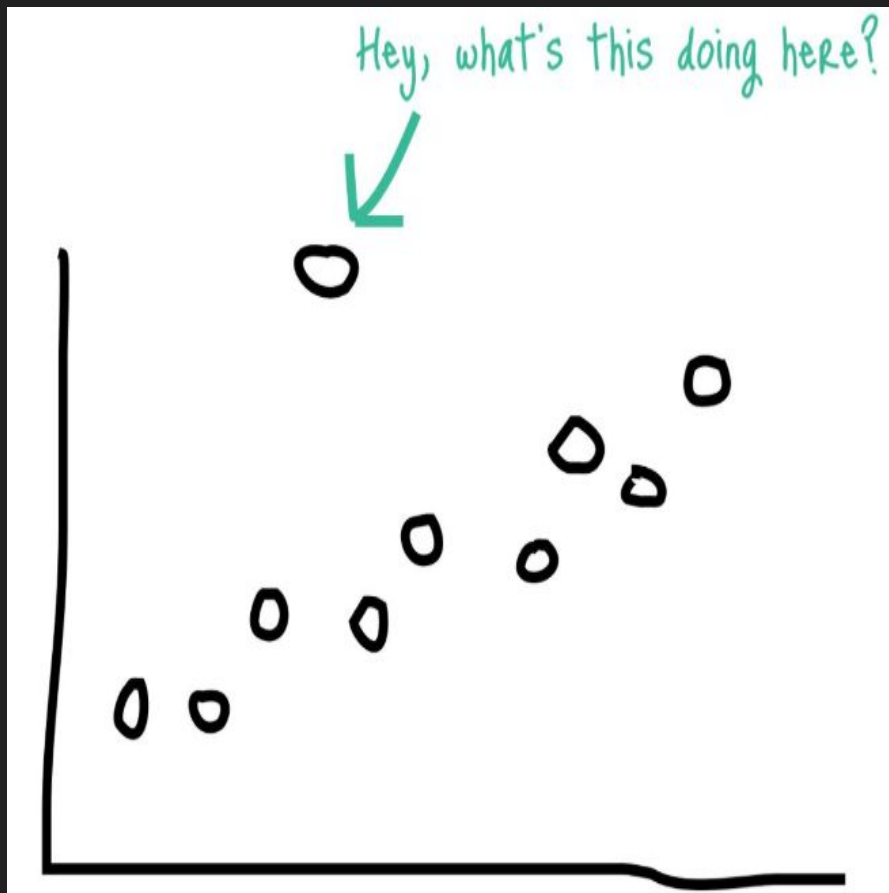
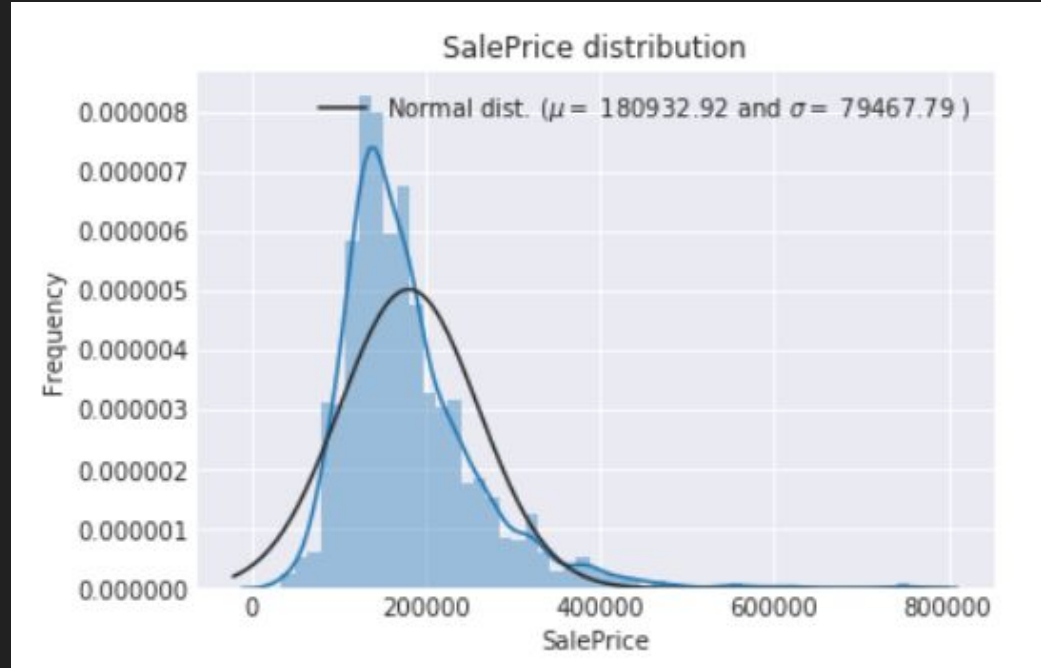# Feature Preprocessing and Engineering

Outliers

Their inclusion is up to the

data scientist, but it is recommended

to have a separate model

for them

Multiplicative Distribution

It is better to transform

multiplicative distribution

features as there are intervals
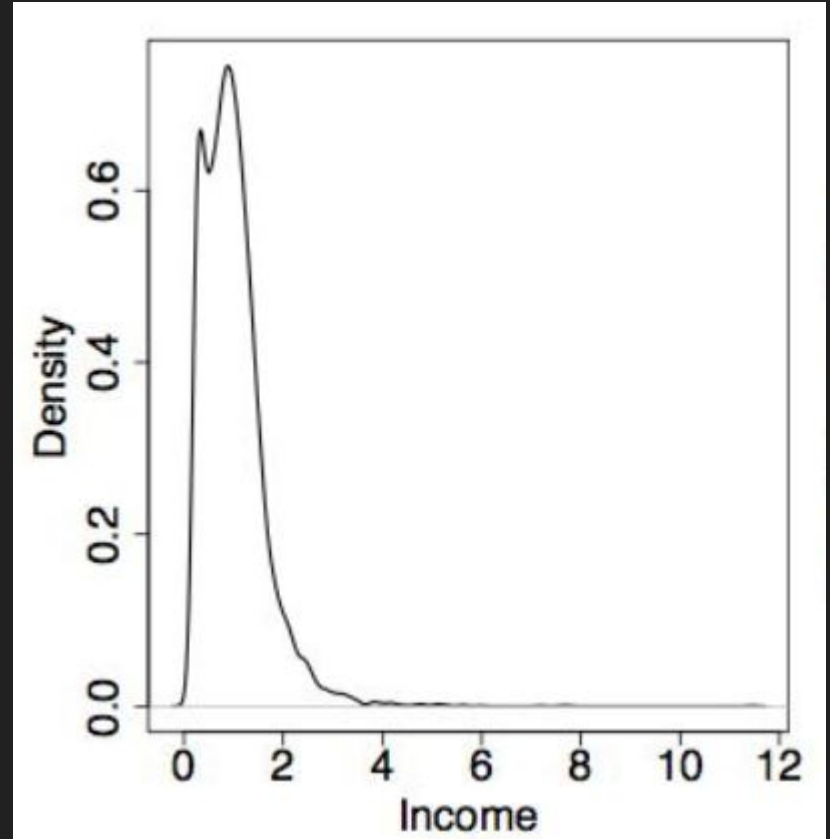
with low number of samples

Skewed Distribution Variables

Skewed Distribution

can complicate the analysis

for some ML models

Imputation

Imputation is the process of replacing missing data.

Some strategies:

- Interpolation (Time Series): Taking the value of nearest non-missing and calculate average
- New Category or Value: Missing is a category
- Most Common: Replace with the most common
- Most common from category: Taking the nearest per another category

Scaling

Different features might have different units and orders of magnitude. Scaling puts all of them in the same order of magnitude.

Examples of Scaling:

- Min-Max scaling
- Standard Scaling
- Robust Scaling

Categorical Features

Categorical features can not be feed directly to a machine learning model. They need to be converted to a number of vector.

Some strategies:

- One-hot encoding: We add a column per feature category
- Weight: We replace the category by a number (sum of target class or mean of target)
- Embedding: We replace the category with a vector
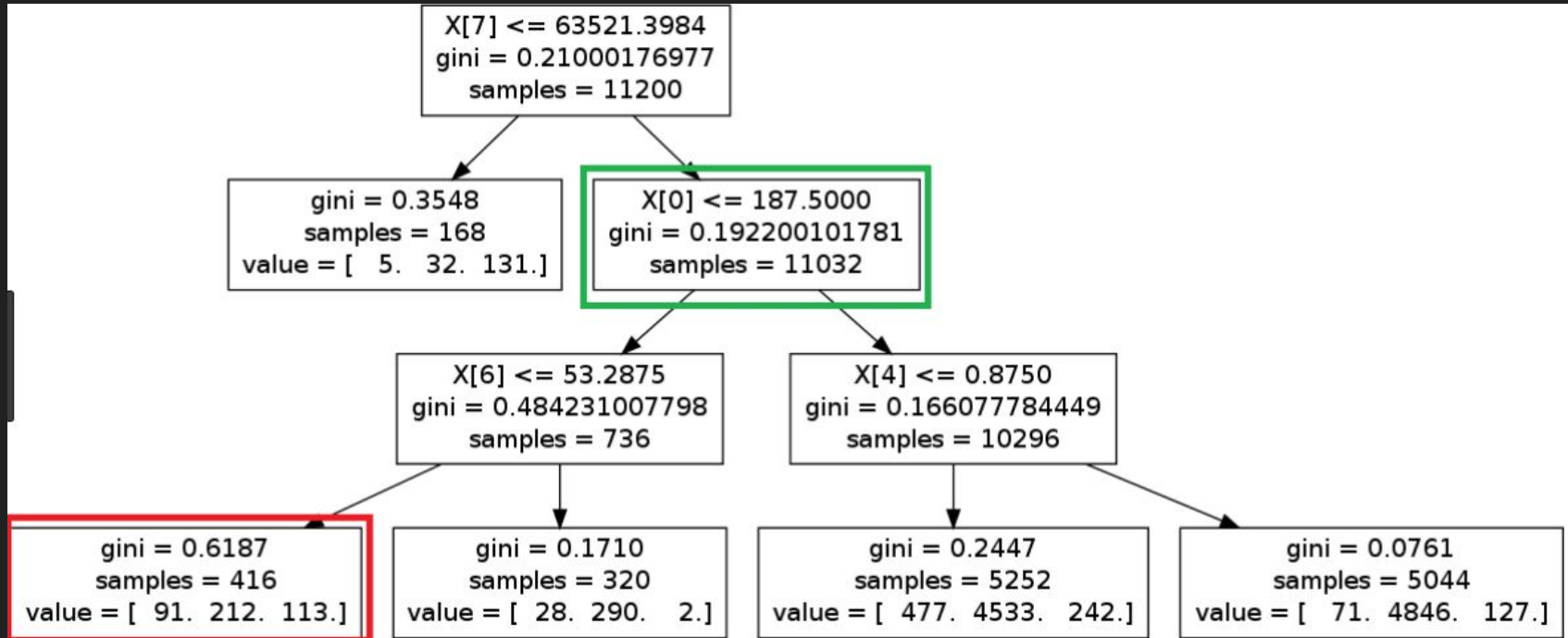
Machine Learning Pipeline

Preprocessing and Machine Learning Models can be integrated in a single model.

This model is called a pipeline.
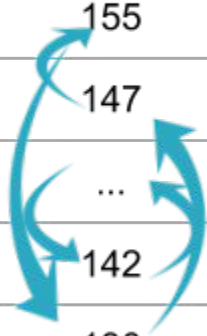
Model Interpretability

Tree Based Models

Variables in the tree are of higher importance and prioritized.

Permutation Importance
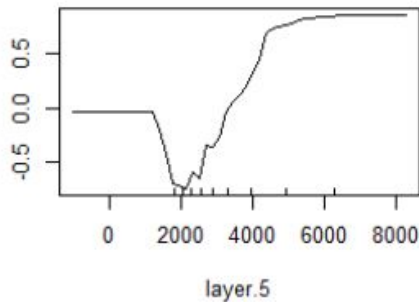
Study of the effect of **switching** values of a feature.

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

Partial Dependence Plot

Effect of the variable on the target
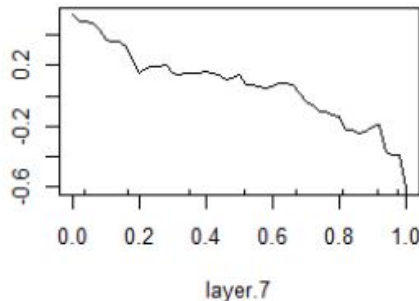
if all the other features were kept

the same
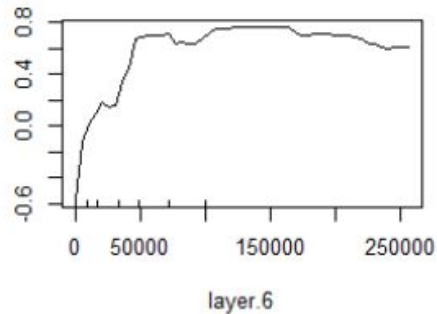
Shap Values
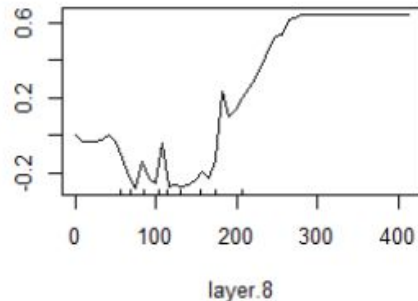
Shap values provide the effect of the different feature for a specific prediction:

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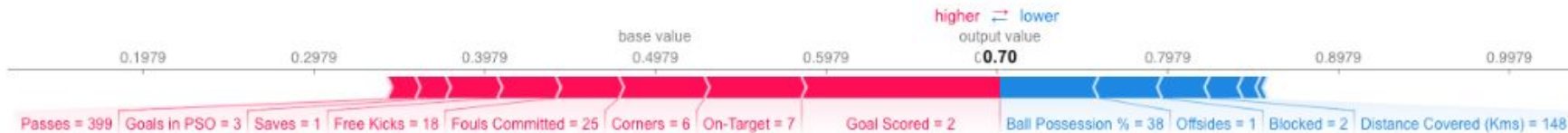
AutoML

AutoML

Several companies/libraries offer the promise of an automated machine learning modelling:

- Massively try different models and approach
- Tweaks the parameters to get to the best result

BUT all those approaches can not replace Data Scientist expertise especially on feature engineering.

AutoML Examples

Companies:

      Data Robot

      Dataiku

Libraries:

      AutoML (for Sklearn)

      TPOT (Genetic Programming)

References

https://towardsdatascience.com/interpretable-machine-learning-1dec0f2f3e6b

https://www.kaggle.com/cast42/lightgbm-model-explained-by-shap

https://towardsdatascience.com/interpretable-machine-learning-1dec0f2f3e6b

https://github.com/slundberg/shap

http://www.f1-predictor.com/model-interpretability-with-shap/

https://christophm.github.io/interpretable-ml-book/

References

http://onlinestatbook.com/2/transformations/box-cox.html

https://en.wikipedia.org/wiki/Imputation_(statistics)

https://towardsdatascience.com/one-feature-attribution-method-to-supposedly-rule-them-all-shapley-values-f3e04534983d

https://romannurik.github.io/SlidesCodeHighlighter/