Mikhail Hushchyn

Feature Engineering

Feature engineering, importance and selection

2021

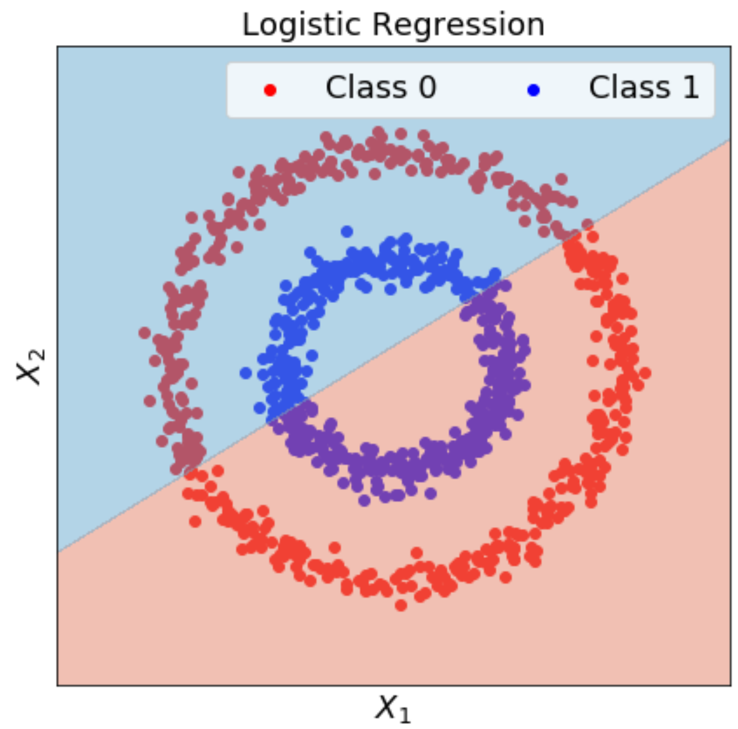
Outline

- Feature engineering
- Feature importance
- Feature selection

Feature Engineering

Example 1

- Consider a binary classification problem in 2D with a Logistic Regression classifier
- Classes are concentric circles and the classifier can not separate them using only features X_1 and X_2

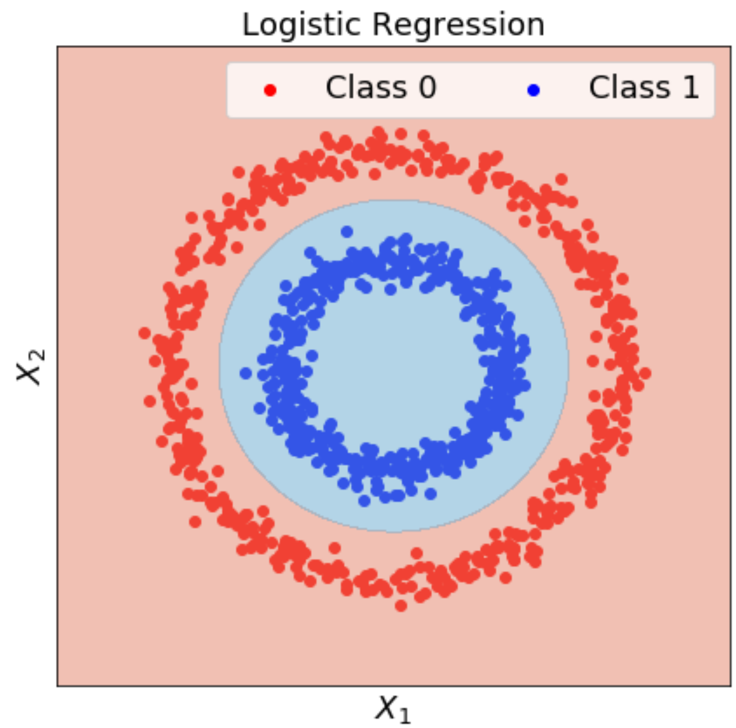


Example 1

- Let's create the new feature X_3:

$$X_3 = X_1^2 + X_2^2$$

- This feature helps to separate the circles by a straight line
- Now, Logistic Regression can solve the classification problem ideally using all three features: X_1, X_2 and X_3

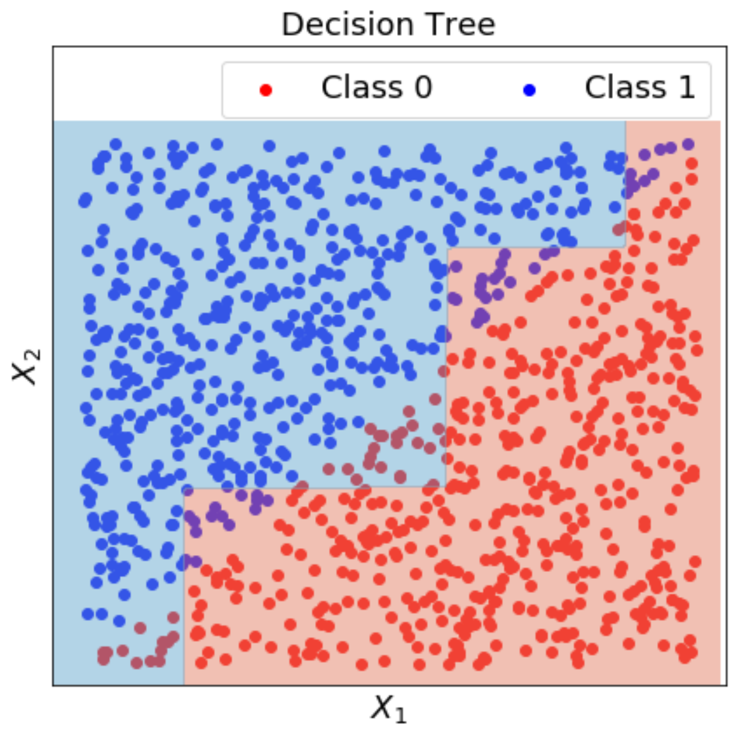


Example 2

- Consider an example where two classes are separated by the surface:

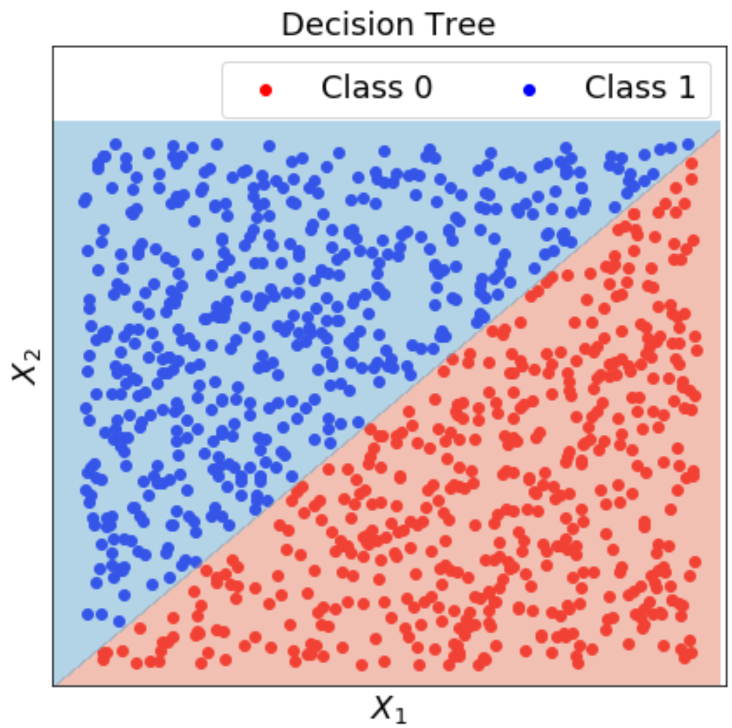
$$X_2 - X_1 = 0$$

- Such surfaces are difficult for Decision Tree classifier
- It requires larger depth of the tree to separate the classes properly



Example 2

- The new feature X_3 helps the classifier:
$$X_3 = X_2 - X_1$$
- Now, the classes are separated using just one predicate $X_3 > 0$
- It requires a Decision Tree with depth = 1 to solve the problem ideally



Advantages of FE

Creating new features can:

- Improve quality of a model
- Reduce complexity of a model (shorter decision trees)
- Speed up model training
- Reduce dimensionality of a problem by removing less informative features (X_1, X_2 in previous examples)

Principles

Key principals in feature engineering:

- Use any information about a problem (classes are circles)
- Create features with physical meaning ($\sqrt{X_1^2 + X_2^2}$ is radius)
- Remove limitations of a model (like with decision tree in example 2)

Typical examples

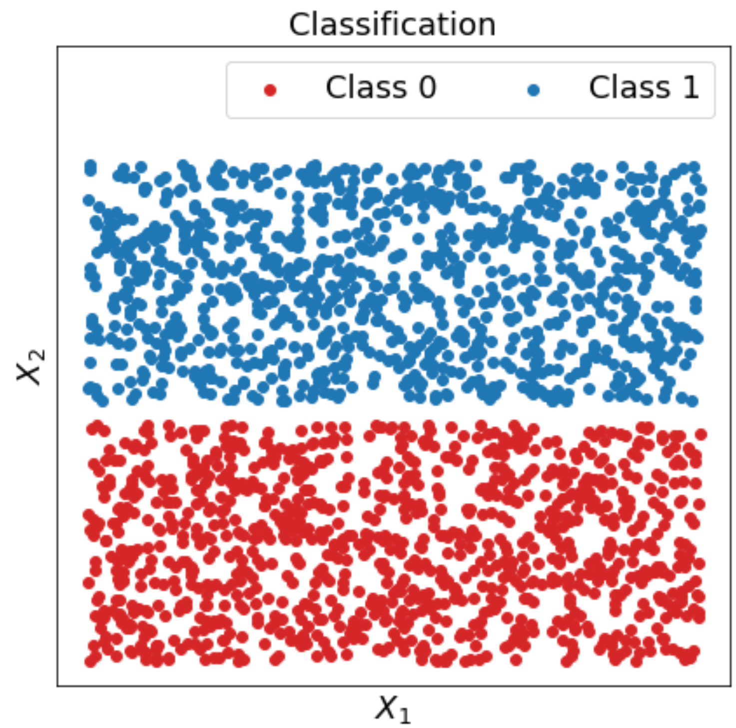
The commonly used feature combinations:

- X_i^p
- $X_1 X_2$
- $X_1^2 \pm X_2^2$
- $X_1 \pm X_2$
- $\frac{X_1 \pm X_2}{X_1 \mp X_2}$
- $\sin X_1, \cos X_1$

Feature Importance

Intuition

- Not all features are equally useful for a problem
- Some of them are more informative than others
- In the example, X_1 is uninformative for the classification problem
- The goal is to measure importance of each feature



Methods

The main feature importance estimation methods:

- Using correlation
- Using probabilistic distance
- Decision tree based
- Linear model based
- General method

Correlation

For a feature f calculate correlation with target y:

$$\rho(f, y) = \frac{\sum_i (f_i - \bar{f})(y_i - \bar{y})}{\sqrt{\sum_i \left(f_i - \bar{f}\right)^2 \sum_i (y_i - \bar{y})^2}}$$

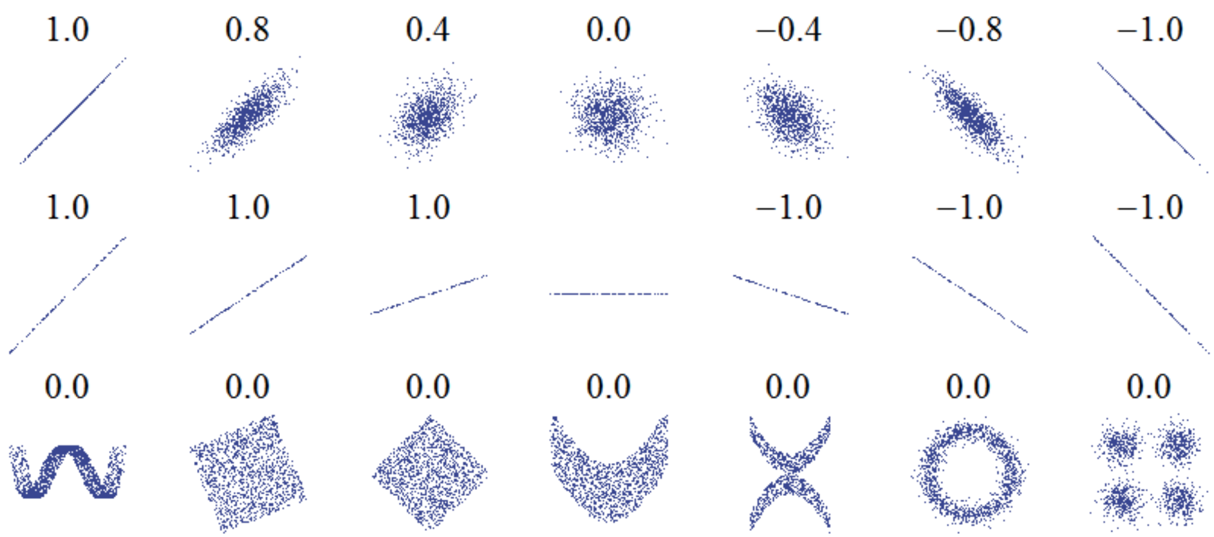
y_i - labels in binary classification or target in regression for the i-th object

f_i - the feature value for the i-th object

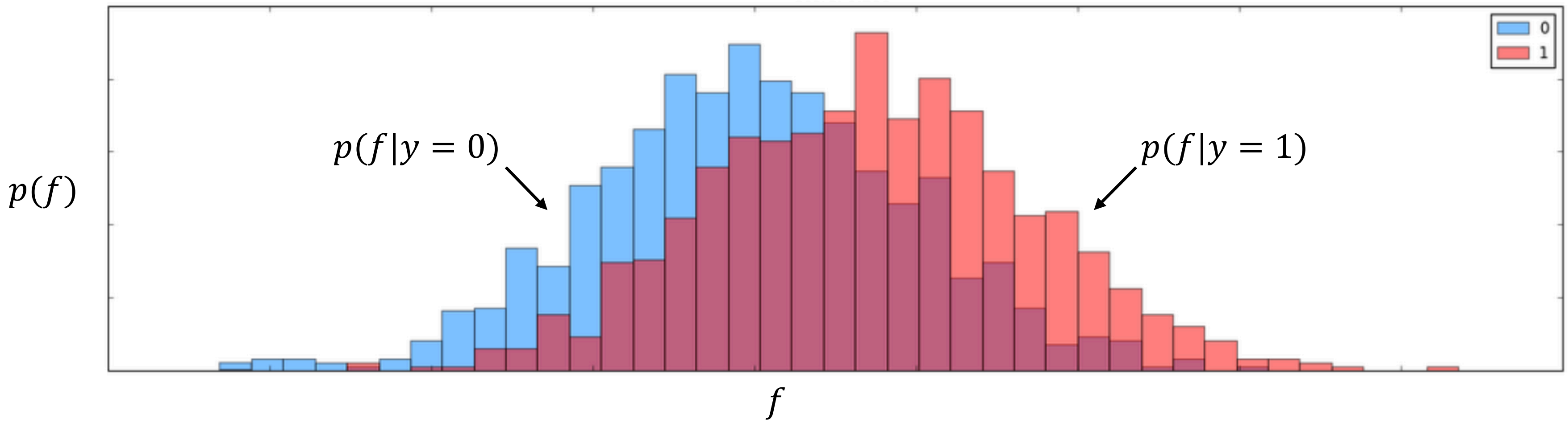
$\rho(f, y)$ - the feature importance $Imp(f)$

Correlation

- Easy to compute
- But captures only linear dependencies



Probabilistic distance



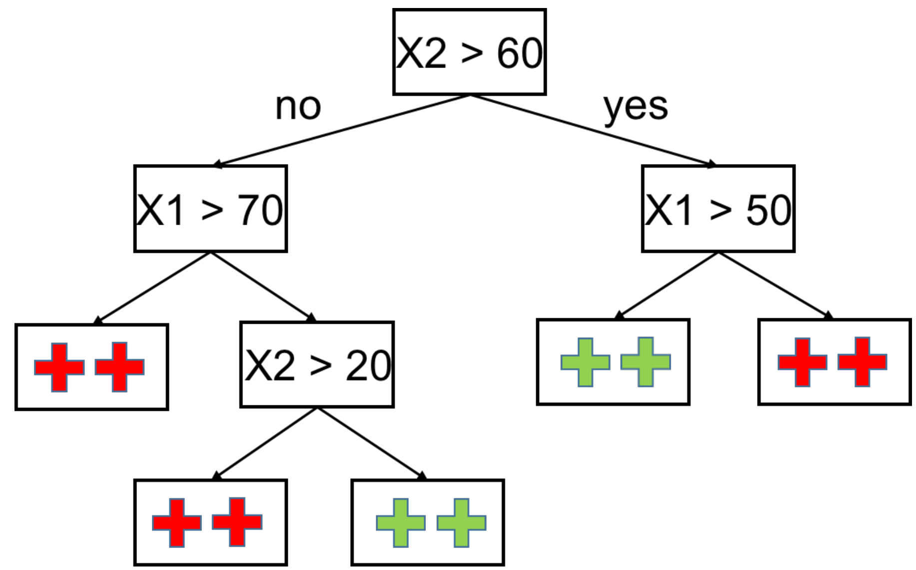
Feature importance $Imp(f)$ as total variation (distance) between two distributions:

$$Imp(f) = \int |p(f|y=1) - p(f|y=0)|df$$

Decision tree based

Decision tree recap:

- Each node t has two children
- n_t - the number of objects in this node
- $I(t)$ – impurity function (gini, cross-entropy, MSE) value for the node



Decision tree based

Let $T(f)$ be the set of all nodes which use feature f to make a split. Then, feature importance of f:

$$Imp(f) = \sum_{t \in T(f)} n_t \Delta I(t)$$

$$\Delta I(t) = I(t) - \sum_{c \in children(t)} \frac{n_c}{n_t} I(c)$$

Linear model based

Consider a linear model with regularization (L_1 or L_2 penalty):

$$\hat{y} = w_0 + w_1 f_1 + w_2 f_2 + \cdots + w_k f_k$$

If features are normalized (have the same ranges), feature importance of f_i is equal to:

$$Imp(f_i) = |w_i|$$

General method

- Train your model
- Calculate quality measure Q_0 on validation set
- For a feature f:
  - Replace given values with random values from the same distribution
  - Or perform random shuffling
  - Calculate quality measure Q_f on validation set
  - Estimate feature importance: $Imp(f) = Q_0 - Q_f$

	RDSpend	Administration	MarketingSpend	Profit	state_California
1	165349.2	136897.8	471784.1	192261.83	0
2	162597.7	151377.59	443898.53	191792.06	1
3	153441.51	101145.55	407934.54	191050.39	1
...	...	...	...	...	...
48	0	135426.92	0	42559.73	1
49	542.05	51743.15	0	35673.41	0
50	0	116983.8	45173.06	14681.4	1

General method (modification)

- Train your model on the full set of features
- Calculate quality measure Q_0 on validation set
- For a feature f:
  - Retrain your model without this feature
  - Calculate quality measure Q_f on validation set
  - Estimate feature importance: $Imp(f) = Q_0 - Q_f$

	RDSpend	Administration	MarketingSpend	Profit	state_California
1	165349.2	136897.8	471784.1	192261.83	0
2	162597.7	151377.59	443898.53	191792.06	1
3	153441.51	101145.55	407934.54	191050.39	1
...	...	...	...	...	...
48	0	135426.92	0	42559.73	1
49	542.05	51743.15	0	35673.41	0
50	0	116983.8	45173.06	14681.4	1

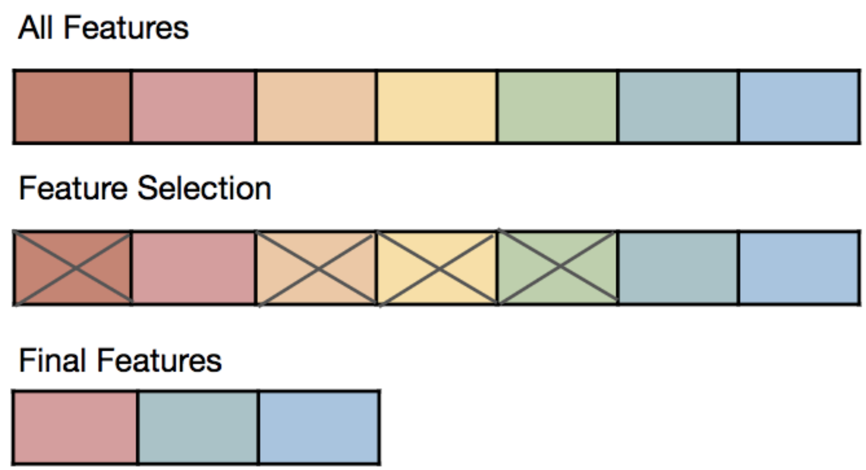
Shapley values

- There are many other approaches to estimate feature importance
- One of the most interesting method is based on Shapley values: https://christophm.github.io/interpretable-ml-book/shapley.html
- With its implementation on python: https://github.com/slundberg/shap

Feature Selection

Feature selection

The goal is to reduce the number of features with minimal loss of model quality.



Examples:

- Keep the best K of D features
- Remove as much features as possible, but keep quality $Q \geq Q_{min}$

Methods

The most popular feature selection approaches:

- Filter method
- Embedded methods
- Recursive feature elimination

Filter method

- Estimate importance for individual features (correlation, probability distance): $Imp(f_1), Imp(f_2), \dots, Imp(f_D)$
- Select the required number of features with the highest importance

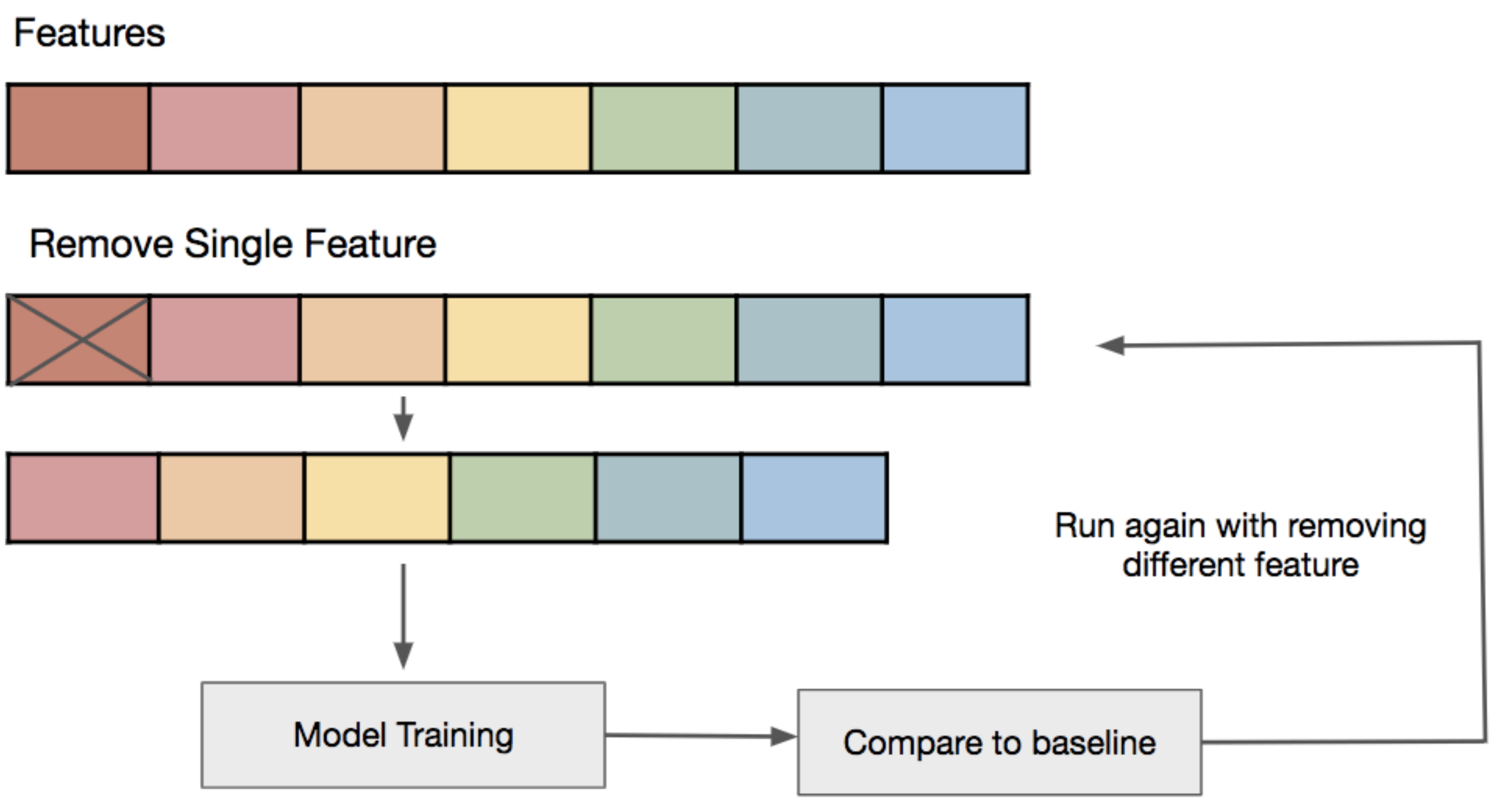
- Simple to implement
- Quite fast
- Bad for correlated features, it takes many redundant ones

Embedded methods

- Based on feature importance of a model
- Linear models:
  - Select the best features using weights of the model (see feature importance section)
  - Use L_1 regularization
- Decision Trees
  - Select the best features using their importance (see feature importance section)

- Widely used
- Takes into account correlations between the features

Recursive feature elimination



Recursive feature elimination

- Train a model on the full set of features
- Estimate feature importance based on the model
- Remove the least important feature or several features
- Repeat

- In combination with the general method for feature importance estimation, this is one of the most powerful methods

Summary

Summary

- Feature engineering
- Feature importance
  - Correlation and probabilistic distance
  - Decision tree and linear model based
  - General method
- Feature selection
  - Filter method
  - Embedded methods
  - Recursive feature elimination