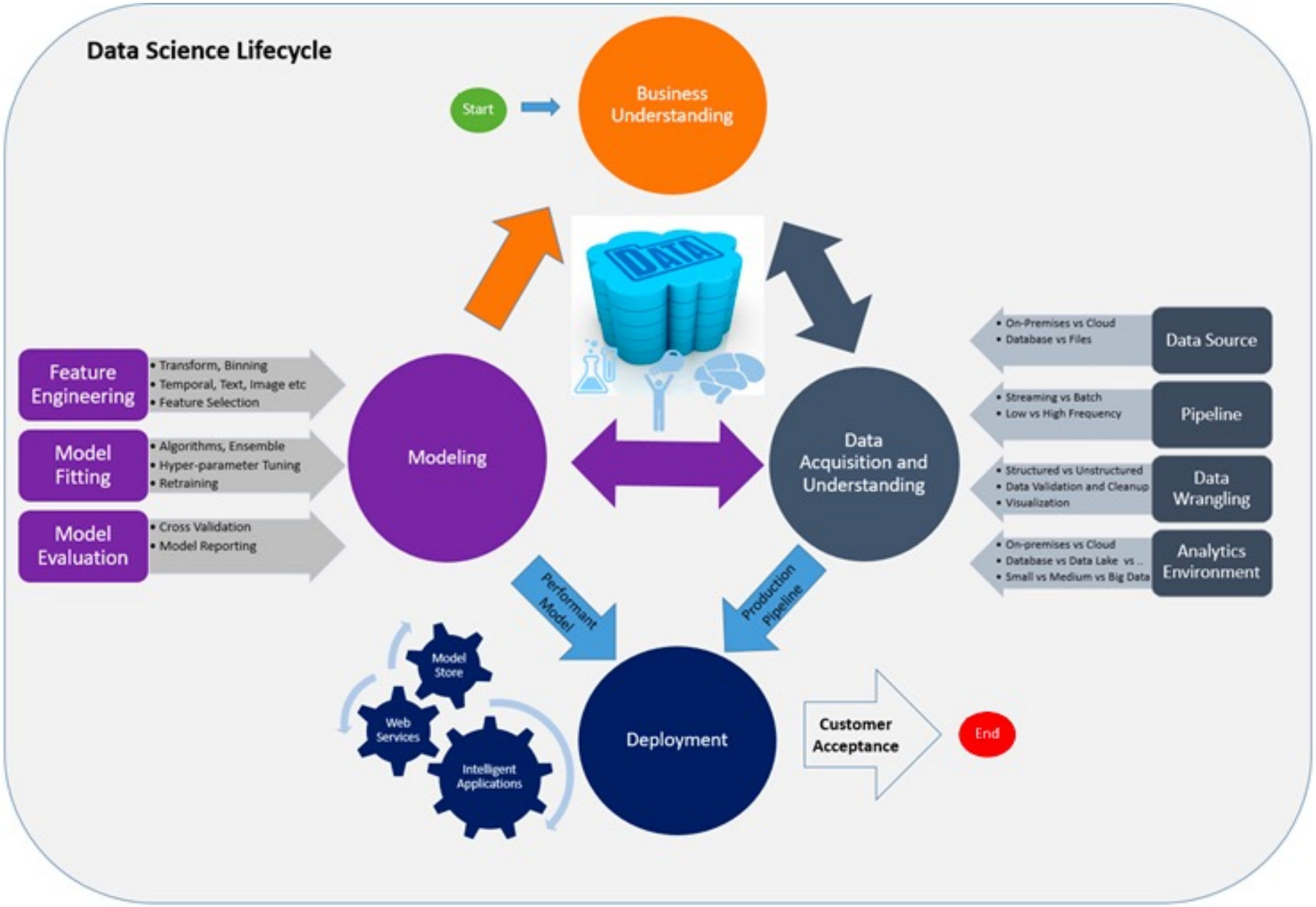
Categorical Data Encoding Techniques. Practical tricks in data analysis tasks

Anastasia Maximovskaya

Data Science Lifecycle



Our plan for today

Feature encoding

Feature generation

Feature selection

Types of features

- Observation (data point, object) – abstract entity, and computers work with numeric
- Feature – numerical characteristic of an object

Lets define the following types:

- Numeric
- Binary (0/1)
- Categorical
- Features with a complex internal structure (images, text)

Feature Encoding

Categorical features

One-Hot Encoding

Human-Readable

Pet
Cat
Dog
Turtle
Fish
Cat

Machine-Readable

Cat	Dog	Turtle	Fish
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

One-Hot Encoding

$$a(x) = w_0 + w_1 \cdot [x_1 = c_1] + w_2 \cdot [x_1 = c_2] + \ldots + w_n \cdot [x_1 = c_n])$$

Ordinal encoding

color
red
green
blue
red

color
0
1
2
0

Target encoding

id	job	job_mean	target
1	Doctor	0,50	1
2	Doctor	0,50	0
3	Doctor	0,50	1
4	Doctor	0,50	0
5	Teacher	1	1
6	Teacher	1	1
7	Engineer	0,50	0
8	Engineer	0,50	1
9	Waiter	1	1
10	Driver	0	0

Feature hashing

- Feature hashing maps each category in a categorical feature to an integer within a pre-determined range

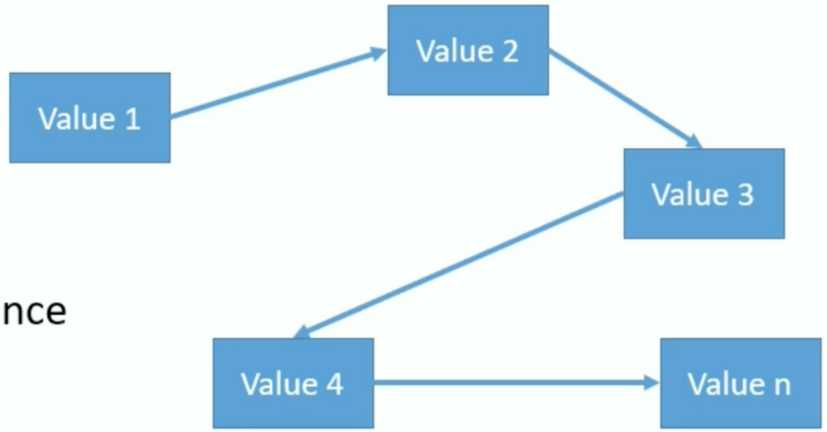
- This output range is smaller than the input range so multiple categories may be mapped to the same integer

- Feature hashing is very similar to one-hot encoding but with a control over the output dimensions.

Feature hashing

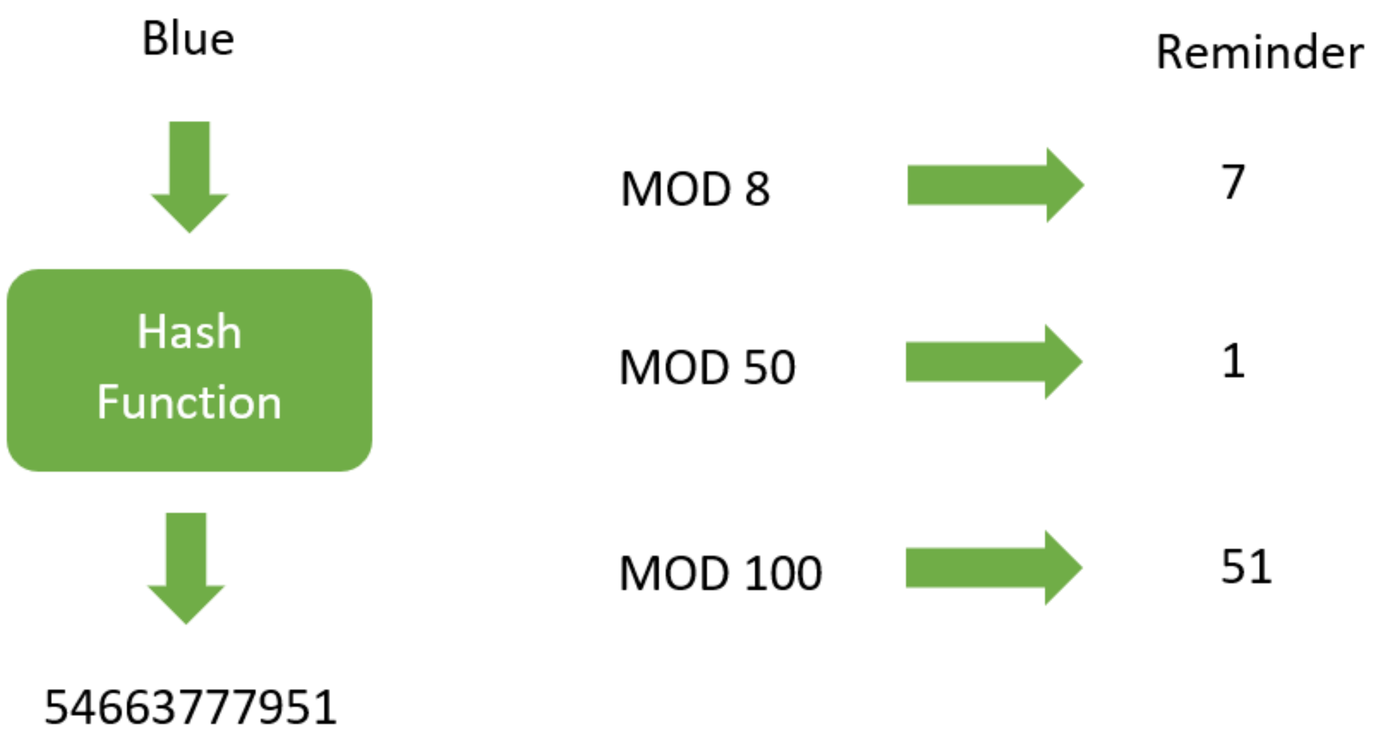
Array size is fixed.
Fixed memory allocation.

Array (3) = value 4



LinkedList can grow dynamically.
Sequential Read hence bad performance

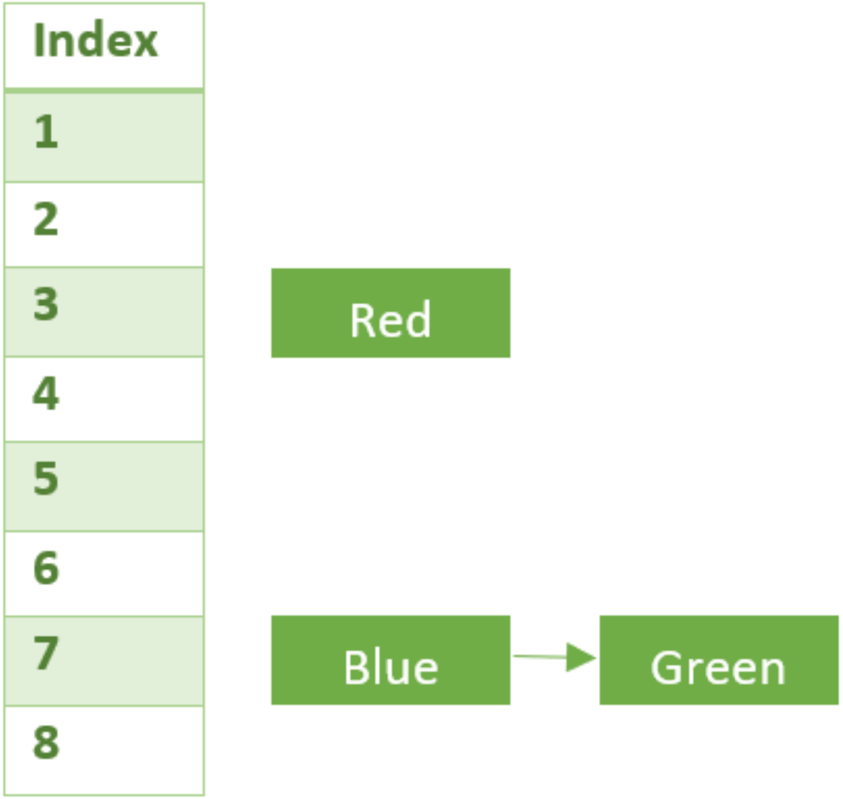
Feature hashing – MOD function



Feature hashing – Hash table

Index	Value
1	
2	
3	"Red"
4	
5	
6	
7	"Blue" – "Green"
8	

Feature hashing – Separate chaining



Feature hashing

Color	Hash Function	Divide by	Reminder
Red	36614357519	8	3
Blue	54663777951	8	7
Green	75535549907	8	7

Feature Hashing

Reminder -->	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Feature 1	**Feature 2**	**Feature 3**	**Feature 4**	**Feature 5**	**Feature 6**	**Feature 7**	**Feature 8**
Red	0	0	0	**1**	0	0	0	0
Blue	0	0	0	0	0	0	0	**1**
Green	0	0	0	0	0	0	0	**1**

Feature hashing

- Implement using category_encoders library
- You can use hash functions from hashlib
- Pro tip: you can find a lot of other ways to encode a categorical feature in this library that did not fit into this lecture ☺

- contrib.scikit-learn.org/category_encoders

Weight of Evidence

- The goal of Weight of Evidence (WOE) is to efficiently identify the best recording to weight-of-evidence values for a list of categorical predictors, and to assign to each category a unique Weight-of-Evidence value

- Weight of Evidence could be used for **combining** variable groups/levels, this process is called **coarse classing**

- We combine categories with similar WOE and then replace the categories with continuous WOE values.

Weight of Evidence

$$WOE = ln\left(\frac{\%\ of\ non - events}{\%\ of\ events}\right)$$

Weight of Evidence

Feature	Outcome
A	1
A	0
A	1
A	1
B	1
B	1
B	0
C	1
C	1

1: event -- 0: non-event

	Non-events	Events	% of Non-events	% of Events	WOE
A	1	3	50	42	$ln\left(\frac{(1+0.5)/2}{(3+0.5)/7}\right) = 0.4$
B	1	2	50	29	$ln\left(\frac{(1+0.5)/2}{(2+0.5)/7}\right) = 0.74$
C	0	2	0	29	$ln\left(\frac{(0+0.5)/2}{(2+0.5)/7}\right) = -0.35$
			100%	100%	

Weight of Evidence

Feature	Outcome	WOE
A	1	0.4
A	0	0.4
A	1	0.4
A	1	0.4
B	1	0.74
B	1	0.74
B	0	0.74
C	1	-0.35
C	1	-0.35

Weight of Evidence – Information Value (IV)

$$IV = \sum(\text{\% of non-events - \% of events}) \times WOE$$

	Non-events	Events	% of Non-events	% of Events	WOE	IV
A	1	3	50	42	$ln\left(\frac{(1+0.5)/2}{(3+0.5)/7}\right) = 0.4$	$(0.5 - 0.42) * 0.4 = 0.032$
B	1	2	50	29	$ln\left(\frac{(1+0.5)/2}{(2+0.5)/7}\right) = 0.74$	$(0.5 - 0.29) * 0.4 = 0.084$
C	0	2	0	29	$ln\left(\frac{(0+0.5)/2}{(2+0.5)/7}\right) = -0.35$	$(0 - 0.29) * -0.35 = 0.105$
			100%	100%		0.221

Weight of Evidence – Rules for IV

Information Value	Variable Predictiveness
Less than 0.02	Not useful for prediction
0.02 to 0.1	Weak predictive Power
0.1 to 0.3	Medium predictive Power
0.3 to 0.5	Strong predictive Power
>0.5	Suspicious Predictive Power

Weight of Evidence

The advantages of WOE transformation are:

- Handles missing values

- Handles categorical variable so there is no need for dummy variables.

- The transformation is based on logarithmic value of distributions. This is aligned with the logistic -regression output function

Feature Encoding

Numeric features

Binning numeric features

$$a(x) = w_0 + w_1 \cdot [t_0 \le x_1 < t_1] + w_2 \cdot [t_1 \le x_2 < t_2] + \ldots + w_n \cdot [t_n \le x_n < t_{n+1}]$$

Binning numeric features

For Example, We have an attribute of age with the following values

Age: 10, 11, 13, 14, 17, 19, 30, 31, 32, 38, 40, 42, 70, 72, 73, 75

Now after Binning, our data becomes:

Attribute	Age -1	Age -2	Age -3
	10, 11, 13, 14, 17, 19	30, 31, 32, 38, 40, 42	70, 72, 73, 75
After Binning	Young	Mature	Old

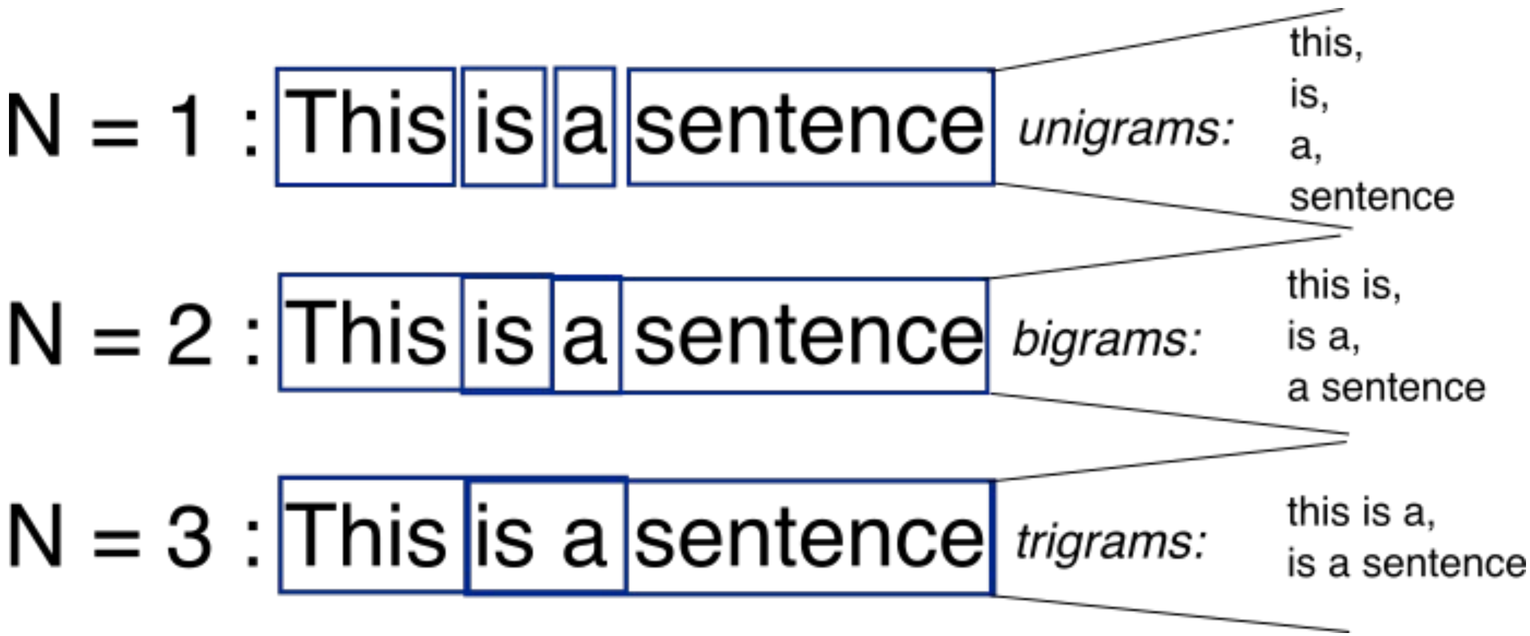
Feature Encoding

Text features

Text preprocessing

- Remove stopwords
- Lemmatization
- Stemming
- Removing punctuation

ngrams



N = 1 : This is a sentence
unigrams:
this,
is,
a,
sentence
N = 2 : This is a sentence
bigrams:
this is,
is a,
a sentence
N = 3 : This is a sentence
trigrams:
this is a,
is a sentence

CountVectorizer

1. the red dog
2. cat eats dog
3. dog eats food
4. red cat eats

the	red	dog	cat	eats	food
1	1	1	0	0	0
0	0	1	1	1	0
0	0	1	0	1	1
0	1	0	1	1	0

TF-IDF Vectorizer

$$\mathrm{tf}(t,d) = \frac{n_t}{\sum_k n_k}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{\, d_i \in D \mid t \in d_i \,\}|}$$

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

Feature Generation

Quick overview

Keep in mind

- Domain specific features
- Task specific features
- General sense

Polynomial Features

- Polynomial features are those features created by raising existing features to an exponent
- For example, if a dataset had one input feature X, then a polynomial feature would be the addition of a new feature (column) where values were calculated by squaring the values in X, e.g. X^2
- This process can be repeated for each input variable in the dataset, creating a transformed version of each

- By generating polynomial features, **we can uncover potential new relationships between the features and the target and improve the model's performance**

Date and time

1. DateTime Components

- Year
- Month
- Week
- Day
- Day of Year
- Day of Week
- Hour
- Minute

2. Boolean Flags

- Is year start
- Is year end
- Is month start
- Is month end
- Is quarter start
- Is quarter end
- Is weekend

3. Time Differences

- Diff in Days
- Diff in Quarters
- Diff in Months
- Diff in Weeks
- Diff in Years

Various aggregations

User	City	Visit Days
1	Roma	1
2	Madrid	2
1	Madrid	1
3	Istanbul	1
2	Istanbul	4
1	Istanbul	3
1	Roma	3

→

User	Istanbul	Madrid	Roma
1	3	1	4
2	4	2	0
3	1	0	0

Feature Selection

Quick overview

Univariate Feature Selection

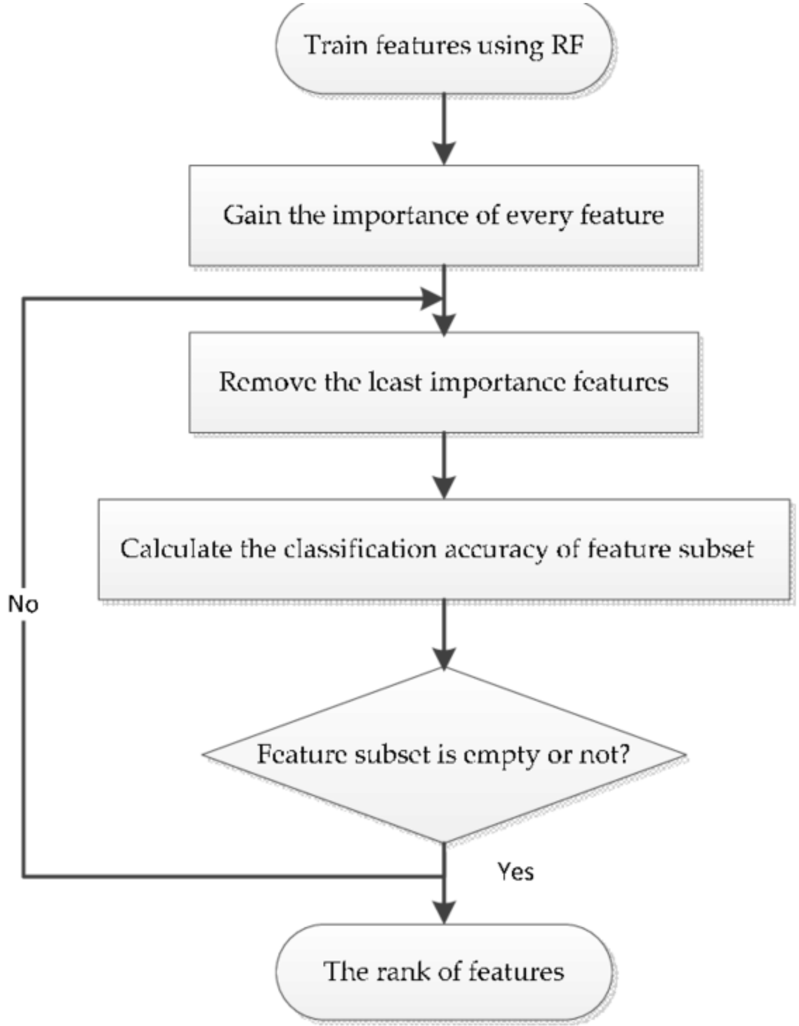
- SelectKBest in sklearn
- Select features according to the k highest scores
- Default is ANOVA F-value, but you can tune it as you see fit

Univariate Feature Selection

Analysis of Variance(ANOVA)

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Squares (MS)	F
Within	$SS_w = \sum_{j=1}^{k}\sum_{j=1}^{l}(X - \overline{X}_j)^2$	$df_w = \mathrm{k} - 1$	$MS_w = \frac{SS_w}{df_w}$	$F = \frac{MS_b}{MS_w}$
Between	$SS_b = \sum_{j=1}^{k}(\overline{X}_j - \overline{X})^2$	$df_b = \mathrm{n} - \mathrm{k}$	$MS_b = \frac{SS_b}{df_b}$	
Total	$SS_t = \sum_{j=1}^{n}(\overline{X}_j - \overline{X})^2$	$df_t = \mathrm{n} - 1$		

Recursive Feature Elimination (RFE)



Model-Based Feature Selection

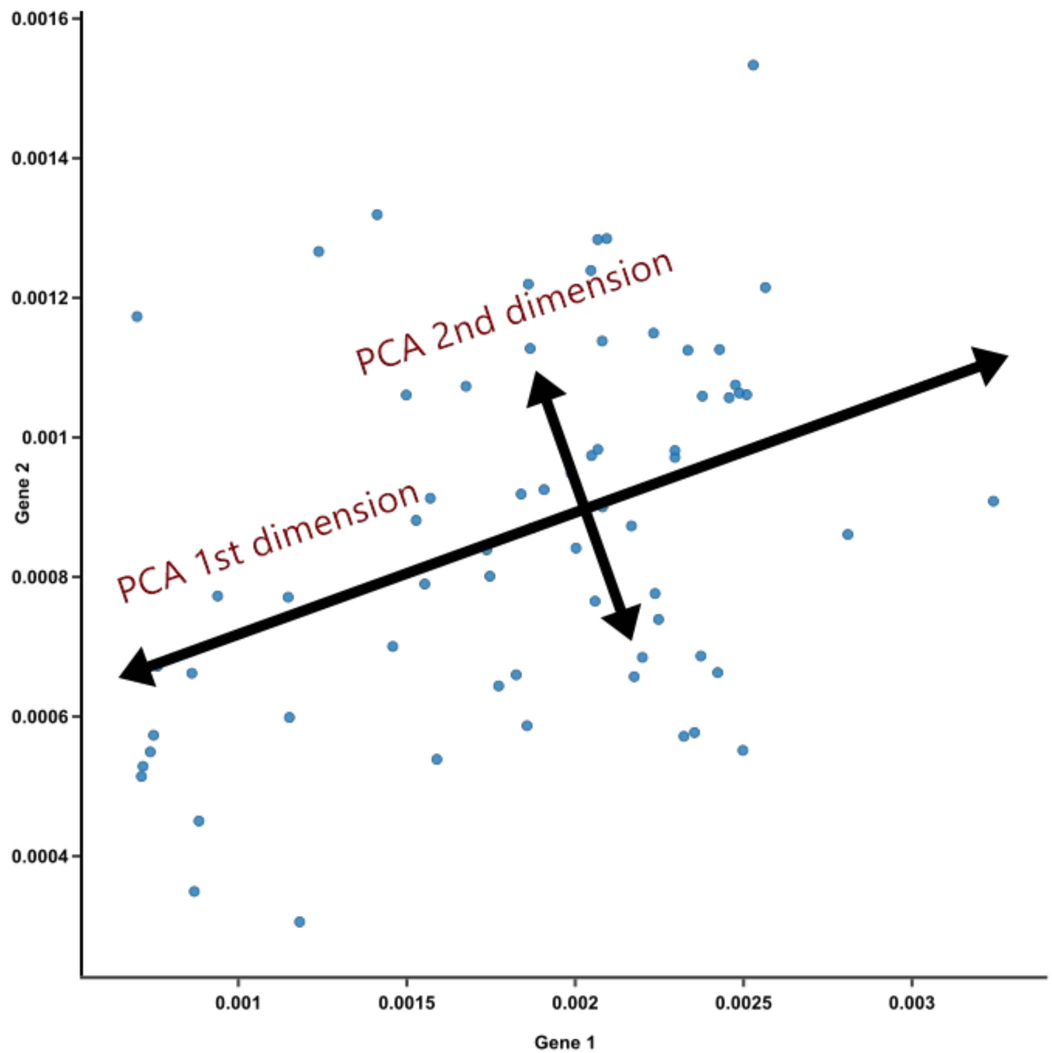
- SelectFromModel in sklearn
- Select features according model’s coefficients or feature importances
- If the attribute value is below the set threshold, these features will be considered unimportant and removed

Model-Based Feature Selection

- Besides specifying a numeric threshold, you can also use the built-in heuristic to find a suitable threshold by specifying a string parameter

- You can use the following heuristics: mean, median, and multiply them by floating point numbers (for example, 0.1 * mean)

Principal Component Analysis



Useful links

- You can check out other winning competitions solutions, for example, here

- More tabular competitions could be found here: zindi

Thank you for your attention!