DATA PREPARATION AND VISUALIZATION

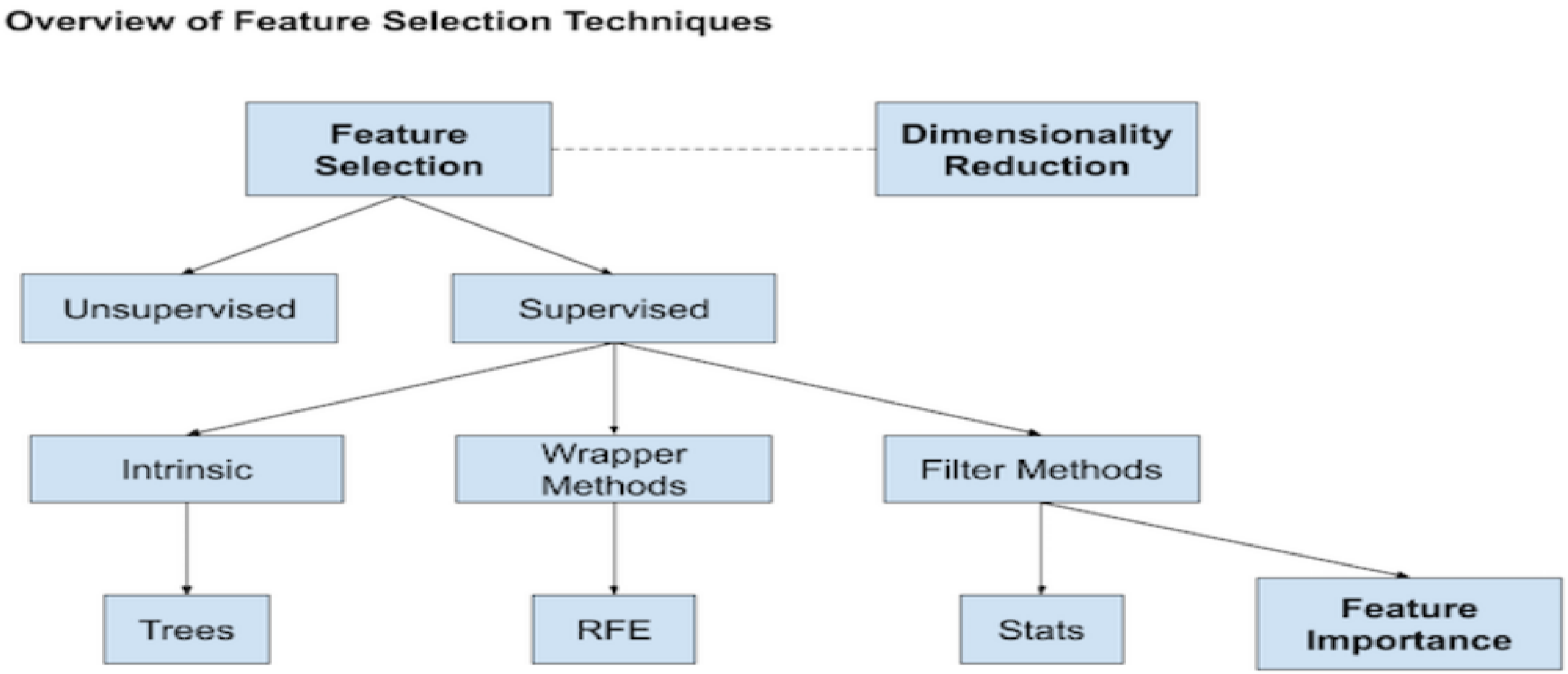
Department of Mathematical Economics

National Economics University
https://www.neu.edu.vn/

Introduction

Feature Selection Techniques

Feature selection refers to techniques for selecting a subset of input features that are most relevant to the target variable that is being predicted



Filter Methods

Filtering

Filtering approach:

Ranks features or features subsets *independently of the predictor*

- Using univariate methods: consider one variable at a time
- Using multivariate methods: consider more than one variables at a time

How to Select Categorical Input Features

- The two most commonly used feature selection methods for categorical input data when the target variable is also categorical (e.g. classification predictive modeling) are the chi-squared statistic and the mutual information statistic

How to Select Categorical Input Features Chi-Squared Feature Selection

- Pearson’s chi-squared statistical hypothesis test is an example of a test for independence between categorical variables
  - Ex:
- The results of this test can be used for feature selection, where those features that are independent of the target variable can be removed from the dataset

Contingency Table Example

Left-Handed vs. Gender

Dominant Hand: Left vs. Right

Gender: Male vs. Female

- 2 categories for each variable, so this is called a 2 x 2 table
- Suppose we examine a sample of 300 children

Contingency Table Example

(continued)

Sample results organized in a contingency table:

sample size = n = 300:

120 Females, 12 were left handed

180 Males, 24 were left handed

Gender	Hand Preference		
	Left	Right	
Female	12	108	120
Male	24	156	180
	36	264	300

χ^2 Test for the Difference Between Two Proportions

$H_0: \pi_1 = \pi_2$ (Proportion of females who are left handed is equal to the proportion of males who are left handed)

$H_1: \pi_1 \neq \pi_2$ (The two proportions are not the same – hand preference is not independent of gender)

- If H_0 is true, then the proportion of left-handed females should be the same as the proportion of left-handed males
- The two proportions above should be the same as the proportion of left-handed people overall

The Chi-Square Test Statistic

The Chi-square test statistic is:

$$\chi^2_{STAT} = \sum_{all\ cells} \frac{(f_o - f_e)^2}{f_e}$$

- where:

  f_o = observed frequency in a particular cell

  f_e = expected frequency in a particular cell if H_0 is true

χ^2_{STAT} **for the 2 x 2 case has 1 degree of freedom**

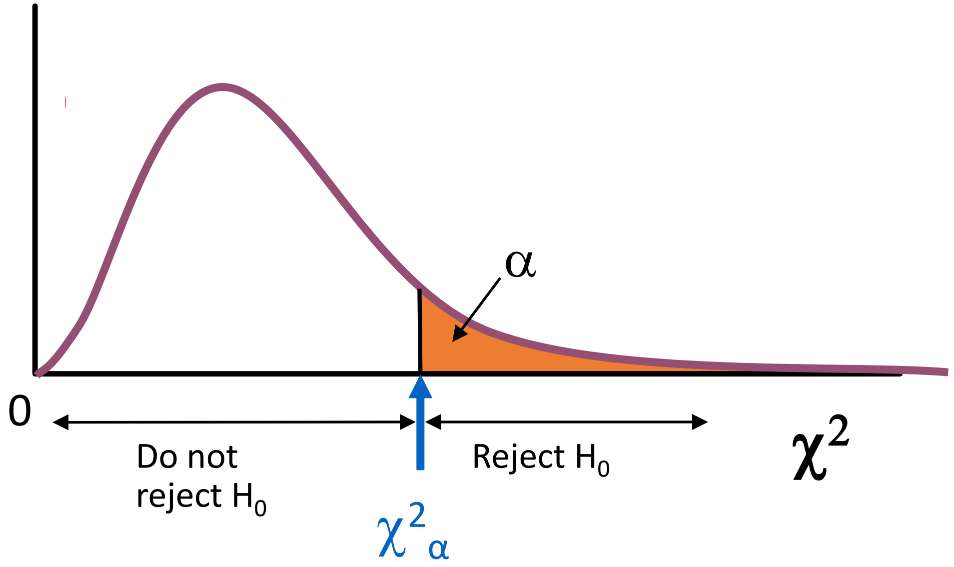
(Assumed: each cell in the contingency table has expected frequency of at least 5)

Decision Rule

The χ^2_{STAT} test statistic approximately follows a chi-squared distribution with one degree of freedom

Decision Rule:

If $\chi^2_{STAT} > \chi^2_{\alpha}$, reject H_0, otherwise, do not reject H_0



Computing the Average Proportion

The average proportion is:

$$\bar{p} = \frac{X_1 + X_2}{n_1 + n_2} = \frac{X}{n}$$

120 Females, 12 were left handed

180 Males, 24 were left handed

Here:

$$\bar{p} = \frac{12 + 24}{120 + 180} = \frac{36}{300} = 0.12$$

i.e., based on all 300 children the proportion of left handers is 0.12, that is, 12%

Finding Expected Frequencies

- To obtain the expected frequency for left handed females, multiply the average proportion left handed ($\bar{p}$) by the total number of females
- To obtain the expected frequency for left handed males, multiply the average proportion left handed ($\bar{p}$) by the total number of males

---

If the two proportions are equal, then

P(Left Handed | Female) = P(Left Handed | Male) = .12

i.e., we would expect (.12)(120) = 14.4 females to be left handed
(.12)(180) = 21.6 males to be left handed

Observed vs. Expected Frequencies

Gender	Hand Preference		
	Left	Right	
Female	Observed = 12 Expected = 14.4	Observed = 108 Expected = 105.6	120
Male	Observed = 24 Expected = 21.6	Observed = 156 Expected = 158.4	180
	36	264	300

The Chi-Square Test Statistic

DCOVA

Gender	Hand Preference		
	Left	Right	
Female	Observed = 12 Expected = 14.4	Observed = 108 Expected = 105.6	120
Male	Observed = 24 Expected = 21.6	Observed = 156 Expected = 158.4	180
	36	264	300

The test statistic is:

$$\chi^2_{STAT} = \sum_{\text{all cells}} \frac{(f_o - f_e)^2}{f_e}$$

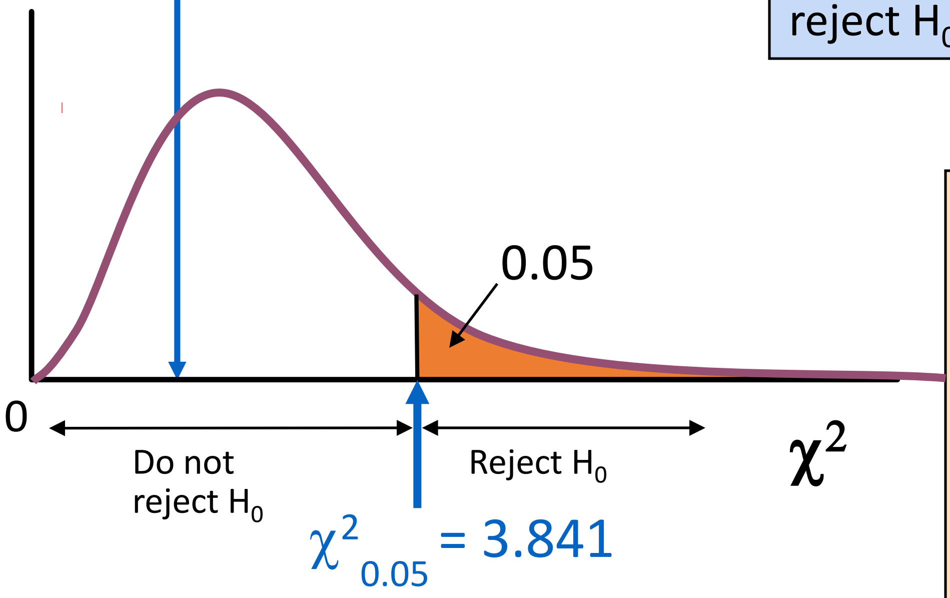
$$= \frac{(12-14.4)^2}{14.4} + \frac{(108-105.6)^2}{105.6} + \frac{(24-21.6)^2}{21.6} + \frac{(156-158.4)^2}{158.4} = 0.7576$$

Decision Rule

The test statistic is $\chi^2_{STAT} = \boxed{0.7576}$; $\chi^2_{0.05}$ with 1 d.f. = 3.841

Decision Rule:

If $\chi^2_{STAT} > 3.841$, reject H_0, otherwise, do not reject H_0



Here,

$\chi^2_{STAT} = 0.7576 < \chi^2_{0.05} = 3.841$,

so we do not reject H_0 and conclude that there is not sufficient evidence that the two proportions are different at $\alpha = 0.05$

How to Select Categorical Input Features
Chi-Squared Feature Selection

- The scikit-learn machine library provides an implementation of the chi-squared test in the chi2() function.
- This function can be used in a feature selection strategy, such as selecting the top *k* most relevant features (largest values) via the SelectKBest class.

How to Select Categorical Input Features
Chi-Squared Feature Selection

Example

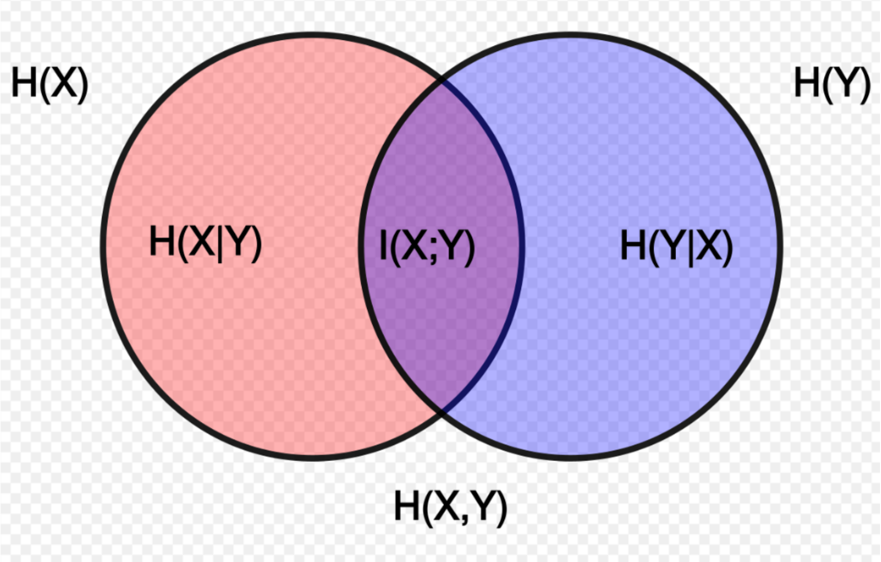
```
>>> from sklearn.datasets import load_digits
>>> from sklearn.feature_selection import SelectKBest, chi2
>>> X, y = load_digits(return_X_y=True)
>>> X.shape
(1797, 64)
>>> X_new = SelectKBest(chi2, k=20).fit_transform(X, y)
>>> X_new.shape
(1797, 20)
```

How to Select Categorical Input Features
Chi-Squared Feature Selection

- The *SelectKBest* class:
  - *The Score_func parameter*: function taking two arrays X and y, and returning a pair of arrays (scores, pvalues). Default is *f_classif*. The default function only works with classification tasks
    - For regression: f_regression, mutual_info_regression
    - For classification: chi2, f_classif, mutual_info_classif
  - *K*: int or “all”, default =10 – number of top features to select
  - Attributes: *scores_* - scores of features

How to Select Categorical Input Features
Mutual Information

- *T*he **mutual information** (**MI**) of two random variables quantifies the "amount of information" obtained about one random variable by observing the other random variable.



Source: Wikipedia

$$\mathrm{H}(X) := -\sum_{x\in\mathcal{X}} p(x)\log p(x) = \mathbb{E}[-\log p(X)]$$

$$\boxed{\mathrm{I}(X;Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P_{(X,Y)}(x,y)\log\left(\frac{P_{(X,Y)}(x,y)}{P_X(x)\,P_Y(y)}\right), \quad \textbf{(Eq.1)}}$$

When x,y independent,p(x,y)=p(x).p(y)=> I(X;Y)=0 => drop this feature

Modeling with Selected Features

- There are many different techniques for scoring features and selecting features based on scores; how do you know which one to use?
- A robust approach is to evaluate models using different feature selection methods (and numbers of features) and select the method that results in a model with the best performance
- Logistic regression is a good model for testing feature selection methods as it can perform better if irrelevant features are removed from the model.

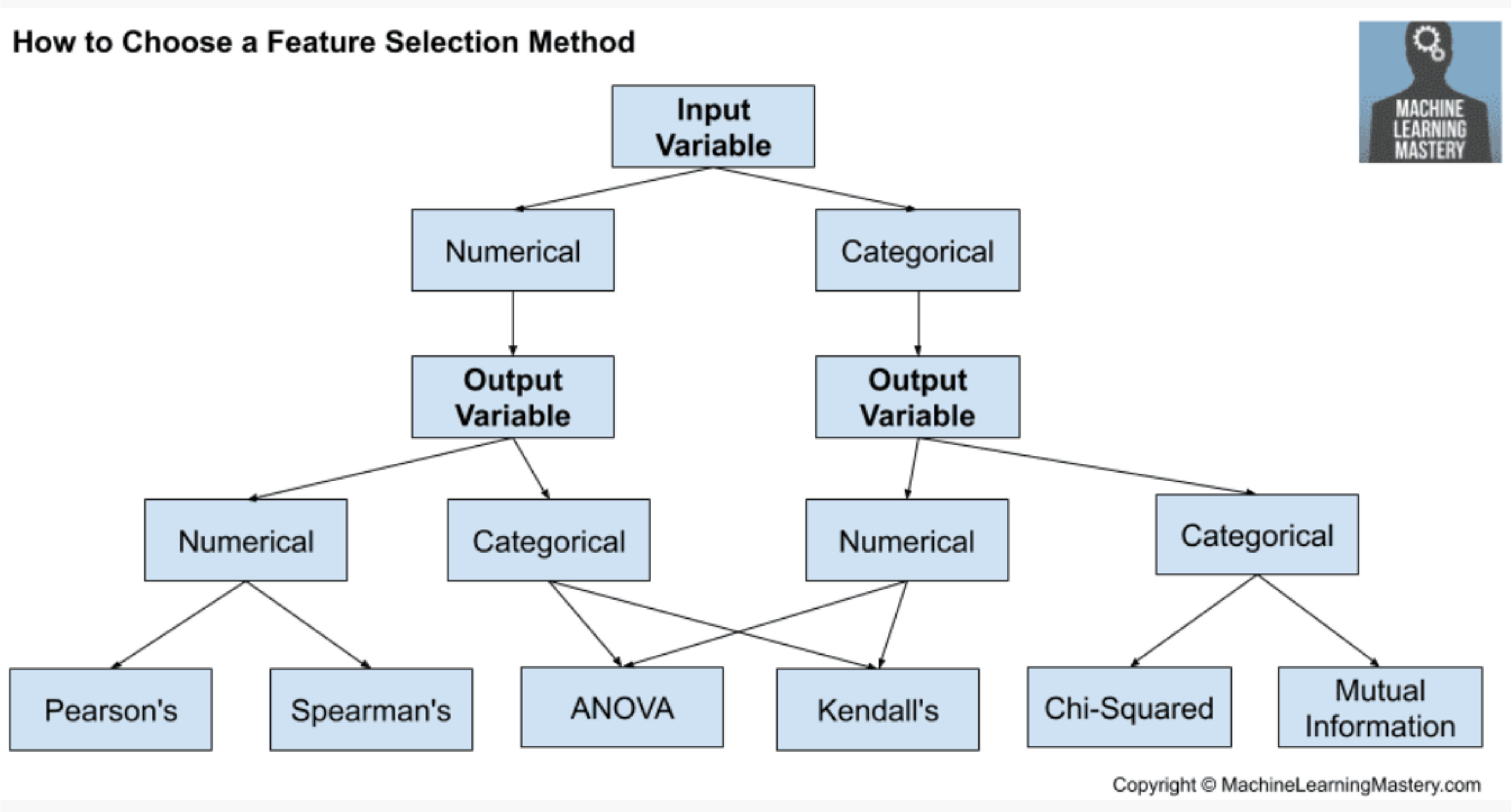
How to select numeric input features

- The two most commonly used features selection methods for numerical input data when target variable is categorical are *the ANOVA F-test statistic and the mutual information statistic*
- ANOVA is used when one variable is numeric and one is categorical
- The results of this test can be used for feature selection where those features that are independent of the target variable can be removed from the dataset

How to Select Features for Numerical Output

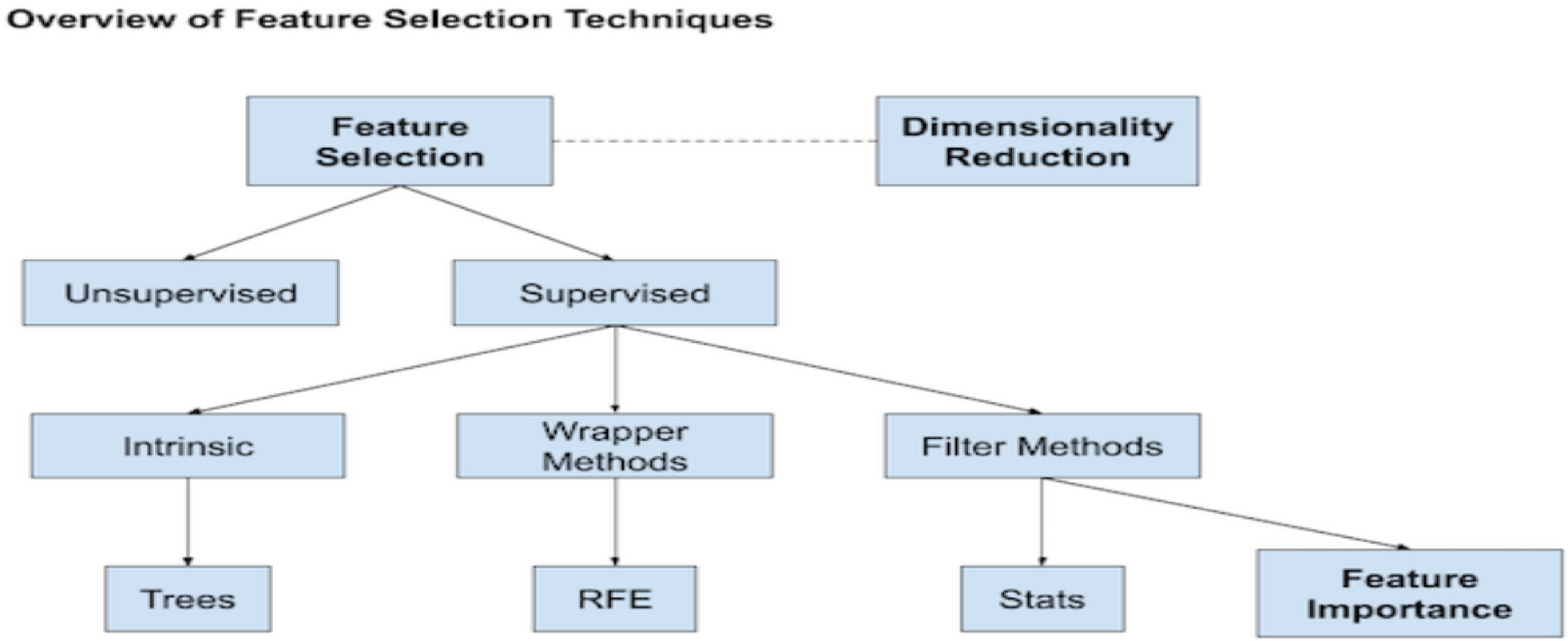
- Correlation is a measure of how two variables change together
- The most common correlation measure is Pearson's correlation that assumes a Gaussian distribution to each variables and reports on their linear relationship
- The scikit-learn machine library provides an implementation of the correlation statistic in the *f_regression()* function

Filter Methods Summary



Wrapper Methods

Wrapper Methods



Wrapper approach: uses *a predictor* to assess (many) features

or feature subsets

Wrapper: Feature Subset Selection

- Two major questions to answer:
  - (a). Assessment: How to asses performance of a learner that uses a particular feature subset?
  - (b). Search: How to search in the space of all feature subsets?

How to Use RFE for Feature Selection

- Recursive Feature Elimination, or RFE for short, is a popular feature selection algorithm.
- There are two important configuration options when using RFE: the choice in the number of features to select and the choice of the algorithm used to help choose features.
-

How to Use RFE for Feature Selection

- RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains
- This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains.

RFE with scikit-learn

sklearn.feature_selection.RFE

class `sklearn.feature_selection.RFE`(*estimator*, ***, *n_features_to_select=None*, *step=1*, *verbose=0*, *importance_getter='auto'*)

[source]

- Parameters:
  - Estimator: a supervised learning estimator with a fit method that provides information about feature importance
  - N_features_to_select: int or float – the number of features to select.
  - Step: int or float, default = 1 – the number of features to remove at each iteration.

RFE with scikit-learn

Example

```
# define dataset
X, y = make_classification(n_samples=1000, n_features=10, n_informative=5, n_redundant=5,
    random_state=1)
# create pipeline
rfe = RFE(estimator=DecisionTreeClassifier(), n_features_to_select=5)
model = DecisionTreeClassifier()
pipeline = Pipeline(steps=[('s',rfe),('m',model)])
# evaluate model
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
n_scores = cross_val_score(pipeline, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
# report performance
print('Accuracy: %.3f (%.3f)' % (mean(n_scores), std(n_scores)))
```

Example of evaluating a model for classification with the RFE transform

RFE with scikit-learn

Methods

`decision_function`(X)	Compute the decision function of `X`.
`fit`(X, y, **fit_params)	Fit the RFE model and then the underlying estimator on the selected features.
`fit_transform`(X[, y])	Fit to data, then transform it.
`get_feature_names_out`([input_features])	Mask feature names according to selected features.
`get_params`([deep])	Get parameters for this estimator.
`get_support`([indices])	Get a mask, or integer index, of the features selected.
`inverse_transform`(X)	Reverse the transformation operation.
`predict`(X)	Reduce X to the selected features and predict using the estimator.
`predict_log_proba`(X)	Predict class log-probabilities for X.
`predict_proba`(X)	Predict class probabilities for X.
`score`(X, y, **fit_params)	Reduce X to the selected features and return the score of the estimator.
`set_params`(**params)	Set the parameters of this estimator.
`transform`(X)	Reduce X to the selected features.

RFE Hyperparameters
Explore Number of Features

- It is good practice to test different values

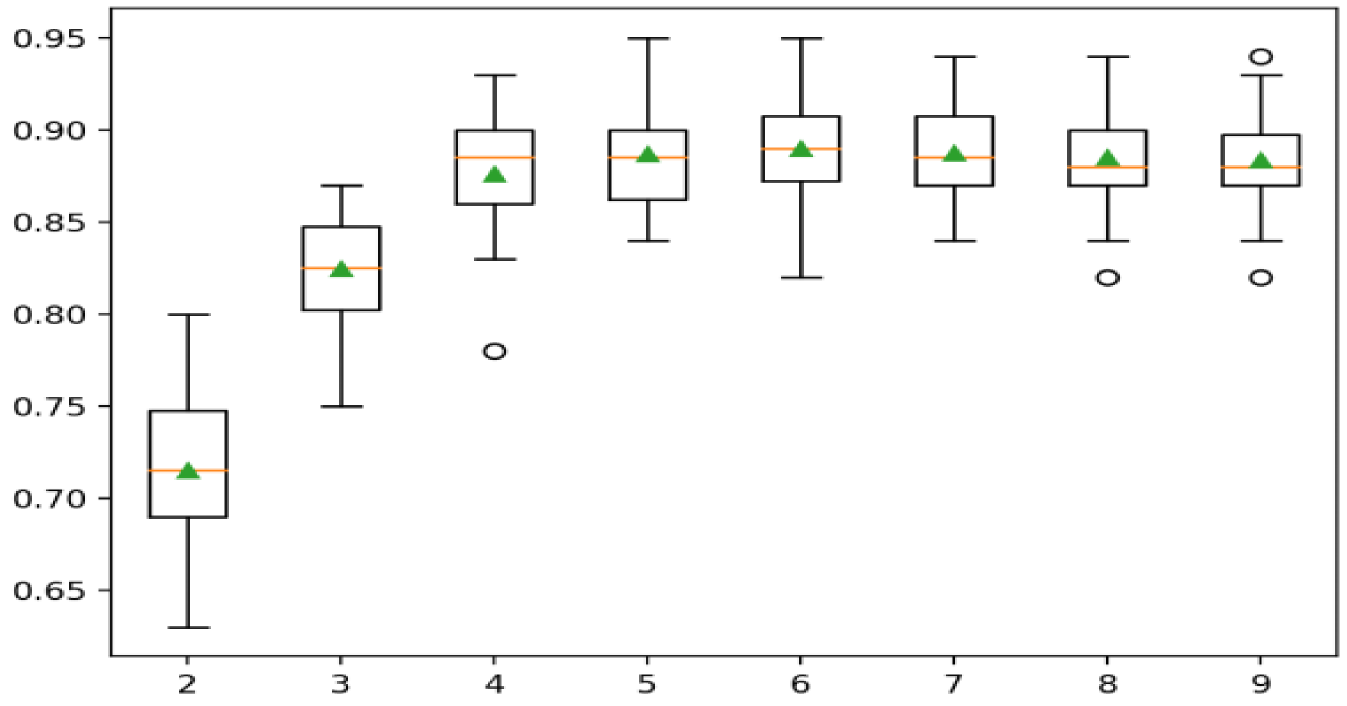


Figure 15.1: Box Plot of RFE Number of Selected Features vs. Classification Accuracy.

RFE – Which Features Were Selected

- When using RFE, we may be interested to know which features were selected and which were removed.
- *The support_ attribute* reports true or false as to which features in order of column index were included
- *The ranking_ attribute* reports the relative ranking of features in the same order