



特征工程



覃雄派



提纲



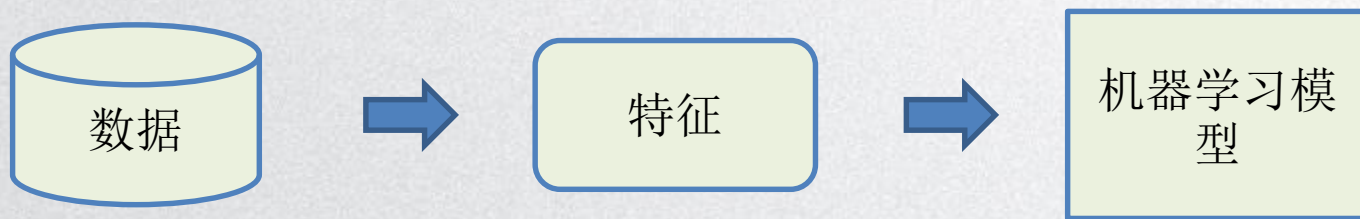
特征工程

- 特征工程入门
 - Feature Transformation
 - Feature Extraction
 - Feature Selection
- 特征工程实践

特征工程

- 特征工程

- Feature Engineering is the way of **extracting/selecting** features from data and **transforming** them into formats that are suitable for Machine Learning algorithms
- 为机器学习模型，准备合适的特征



<https://towardsdatascience.com/feature-engineering-in-python-part-i-the-most-powerful-way-of-dealing-with-data-8e2447e7c69e>

特征工程

特征工程

- 特征工程概览

- 包含3大块

1. Feature Transformation

- 1.1 Categorical - Label Encoding
- 1.2 Categorical - One Hot Encoding
- 1.3 Numeric - Binning
- 1.4 Numeric - Scaling
- 1.5 Log Transform
- 1.6 Handling Outliers
- 1.7 Imputation - Missing Value Handling
- 1.8 Invalid value
- 1.9 Feature Grouping
- 1.10 Feature Split

2. Feature Extraction

- 2.1 降维PCA
- 2.2 Feature Aggregation
-

3. Feature Selection

- 3.1 Filter methods
- 3.2 Wrapper methods
- 3.3 Embedded methods



特征工程

- Feature Engineering is divided into 3 broad categories
 - **(1) Feature Transformation**
 - It means transforming our original feature to the functions of original features
 - Scaling, discretization, binning and filling missing data values are the most common forms of data transformation
 - To reduce right skewness of the data, we can use log
 -



特征工程

- Feature Engineering is divided into 3 broad categories
 - **(2) Feature Extraction**
 - When the data to be processed through an algorithm is too large, it's generally considered redundant
 - Analysis with a large number of variables uses a lot of computation power and memory
 - therefore we should **reduce the dimensionality** of these types of variables
 - It is a term for constructing combinations of the variables
 - For tabular data, we use **PCA** to reduce features
 - For image, we can use line or edge detection



特征工程

- Feature Engineering is divided into 3 broad categories
 - **(3) Feature Selection**
 - All features aren't equal, There are certain features which are more important than other features to the accuracy of the model
 - It is all about selecting a small subset of features from a large pool of features
 - We select **those attributes which best explain the relationship of an independent variable with the target variable**
 - It is **different from dimensionality reduction** because the dimensionality reduction method does so by combining existing attributes, whereas the feature selection method includes or excludes those features
 - The methods of Feature Selection are **Chi-squared test, correlation coefficient scores, LASSO, Ridge regression etc.**

特征工程



特征工程

- 1. Feature Transformation
 - Machine learning algorithms sometimes expect data formatted in a certain way, and that is where feature engineering can help us
 - we need to apply some techniques so our data is compatible with the machine learning algorithm

相对于Feature Extraction
Feature Transformation的处理比较初级



特征工程

- 1. Feature Transformation
 - **Categorical Feature Encoding**
 - Some machine learning algorithms support categorical variables without further manipulation, but some don't
 - That is why we use a **categorical encoding**
- 1.1 Categorical - Label Encoding
 - Label encoding is **converting** each categorical value into some number
 - For example, the "species" feature contains 3 categories. We can assign value 0 to Adelie, 1 to Gentoo and 2 to Chinstrap

特征工程

- 1. Feature Transformation
- 1.2 Categorical - One Hot Encoding
 - It spreads the values in a feature to **multiple flag features** and assigns values 0 or 1 to them
 - This binary value represents the relationship between non-encoded and encoded features
 - For example, in our dataset, we have two possible values in "sex" feature: FEMALE and MALE
 - This technique will create **two separate features** labeled let's say **'FEMALE '** and **'MALE '**

	species	island	sex	Adelie	Chinstrap	Gentoo	Biscoe	Dream	Torgersen	FEMALE	MALE
0	Adelie	Torgersen	MALE	1	0	0	0	0	1	0	1
1	Adelie	Torgersen	FEMALE	1	0	0	0	0	1	1	0
2	Adelie	Torgersen	FEMALE	1	0	0	0	0	1	1	0
3	Adelie	Torgersen	MALE	1	0	0	0	0	1	0	1
4	Adelie	Torgersen	FEMALE	1	0	0	0	0	1	1	0

特征工程

- 1. Feature Transformation
- 1.2+ Categorical - Count Encoding
 - Count encoding is converting each categorical value to its frequency, i.e.. the number of times it **appears** in the dataset

	species	island	sex	species_count_enc	island_count_enc	sex_count_enc
0	Adelie	Torgersen	MALE	152	52	178
1	Adelie	Torgersen	FEMALE	152	52	165
2	Adelie	Torgersen	FEMALE	152	52	165
3	Adelie	Torgersen	MALE	152	52	178
4	Adelie	Torgersen	FEMALE	152	52	165
...
339	Gentoo	Biscoe	MALE	123	167	178
340	Gentoo	Biscoe	FEMALE	123	167	165
341	Gentoo	Biscoe	MALE	123	167	178
342	Gentoo	Biscoe	FEMALE	123	167	165
343	Gentoo	Biscoe	MALE	123	167	178

- Species属性的取值Adelie在样本里出现152次
- 每个样本，如果species属性的取值为Adelie，该列为152

特征工程

- 1. Feature Transformation
- 1.3 Numeric – Binning
 - Binning is a simple technique that **groups** different values into **bins**
 - For example, when we want to **bin numerical features** that would look like something like this:
 - 0-10 → Low
 - 10-50 → Medium
 - 50-100 → High

特征工程

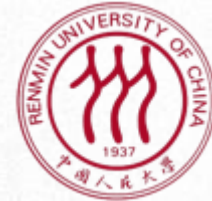
- 1. Feature Transformation

- 1.3 Numeric – Binning

- Binning is a simple technique that **groups** different values into **bins**
- However, we can **bin categorical values too**. For example, we can bin countries by the continent it is on:

- Serbia → Europe
- Germany → Europe
- Japan → Asia
- China → Asia
- USA → North America
- Canada → North America

- The problem with binning is that it can **downgrade performance**
- but it can **prevent overfitting** and increase the robustness of the machine learning model



特征工程

- 1. Feature Transformation
- 1.4 Numeric – Scaling
 - **Scaling** is done for one simple reason, if features are not in the same range, they will be treated **differently** by the machine learning algorithm
 - To put it in lame terms, if we have one feature that has a range of values from 0-10 and another 0-100, a machine learning algorithm might deduce that the second feature is more important than the first one just because it has a higher value
 - Some of machine learning algorithms even require that features look like standard normally distributed data

特征工程

- 1. Feature Transformation
- 1.4 Numeric – Scaling
 - (1) Standard Scaling
 - This type of scaling **removes** mean and scale data to unit variance. It is defined by the formula

$$x_{scaled} = (x - mean) / std$$

- where **mean** is the mean of the training samples, and **std** is the standard deviation of the training samples
- We can use *StandardScaler* class of *Scikit Learn* library

特征工程

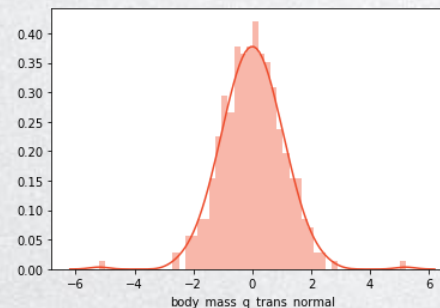
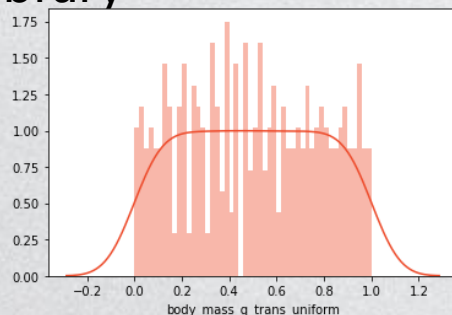
- 1. Feature Transformation
- 1.4 Numeric – Scaling
 - (2) Min-Max Scaling (Normalization)
 - The most popular scaling technique is **normalization** (also called *min-max normalization* and *min-max scaling*)
 - It scales all data in the 0 to 1 range. This technique is defined by the formula

$$X_{std} = (X - X.min(axis = 0)) / (X.max(axis = 0) - X.min(axis = 0))$$
$$X_{scaled} = X_{std} * (max - min) + min$$

- We can use *MinMaxScaler* of *Scikit learn* library

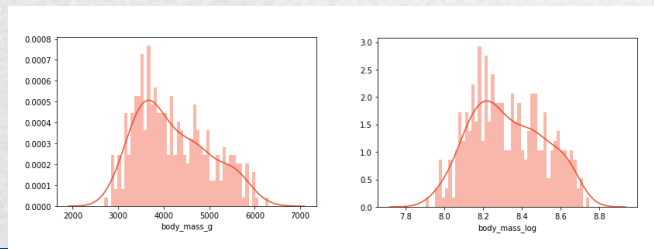
特征工程

- 1. Feature Transformation
- 1.4 Numeric – Scaling
 - (3) Quantile Transformation
 - sometimes machine learning algorithms **require** that the distribution of our data is **uniform** or **normal**
 - here is how it looks like when we transform our data to uniform distribution & normal distribution
 - We can achieve that using *Quantile Transformer* class of *Scikit Learn* library



特征工程

- 1. Feature Transformation
- 1.5 Log Transform
 - This transformation brings many **benefits**. One of them is that the distribution of the data becomes **more normal**. In turn, this helps us to **handle skewed data** and decreases the impact of the **outliers**
 - It is important to note that data must be **positive**, so if you need a scale or normalize data beforehand
 - If we check the distribution of non-transformed data and transformed data we can see that transformed data is closer to the normal distribution



特征工程

- 1. Feature Transformation

- 1.6 Handling Outliers

- Outliers are values that are deviating from the whole **distribution** of the data
- Sometimes these values are mistakes and wrong measurements and should be removed from datasets, but sometimes they are valuable **edge-case** information

- In a nutshell, we can use the **Inter-quartile range** to detect these points
- Data between $Q1$ and $Q3$ is the IQR . Outliers are defined as samples that fall below $Q1 - 1.5(IQR)$ or above $Q3 + 1.5(IQR)$. We can do this using a **boxplot**

特征工程

- 1. Feature Transformation

- 1.6 Handling Outliers

- Outliers are values that are deviating from the whole **distribution** of the data
- Sometimes these values are mistakes and wrong measurements and should be removed from datasets, but sometimes they are valuable **edge-case** information

- The other way for detecting and removing outliers would be by using standard deviation
- Here we need to define the **factor** by which we multiply the standard deviation. Usually, we use values between 2 and 4 for this purpose
- Data in $[\text{mean} - \text{factor} * \text{standard deviation}, \text{mean} + \text{factor} * \text{standard deviation}]$ is normal, otherwise is outlier

特征工程

- 1. Feature Transformation

- 1.6 Handling Outliers

- Outliers are values that are deviating from the whole **distribution** of the data
- Sometimes these values are mistakes and wrong measurements and should be removed from datasets, but sometimes they are valuable **edge-case** information

- Finally, we can use a method to detect outliers is to use **percentiles**
- We can assume a certain percentage of the value from the top or the bottom as an outlier



特征工程

- 1. Feature Transformation
- 1.7 Imputation - Missing Value Handling
 - We can **Drop** the whole sample
 - Or we can **replace** missing values
 - Numeric feature: mean均值, mode众数, median中位数
 - Categorical feature: **most frequent value**



特征工程

- 1. Feature Transformation
- 1.8 Invalid value
 - **invalid** value is different from missing value
 - For the sex feature, the value should be MALE or Female
 - If the value is ".", then it is invalid
 - **Drop** the whole sample
 - or **replace** missing value

特征工程

- 1. Feature Transformation

- 1.9 Feature Grouping

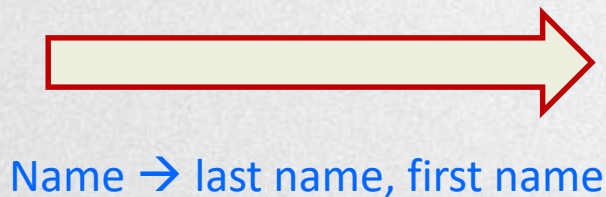
- sometimes we have observations that are **spread** over several rows. The goal of the *Feature Grouping* is to connect these rows into a single one and then use those **aggregated rows**
- The main question when doing so is which type of aggregation function will be applied to features. This is especially complicated for categorical features
 - Here we grouped data by *species* value and for each numerical value we created two new features with sum and mean value

	culmen_length_mm_sum	culmen_depth_mm_sum	culmen_length_mm_mean	culmen_depth_mm_mean
species				
0	5901.42193	2787.45117	38.825144	18.338495
1	3320.70000	1252.60000	48.833824	18.420588
2	5842.52193	1844.25117	47.500178	14.993912

特征工程

- 1. Feature Transformation
- 1.10 Feature Split
 - Sometimes, data is not connected over rows, but over **columns**
 - if we want to extract only first name from this feature
 - This technique is called feature splitting and it is often used with string data

data.names	
0	Andjela Zivkovic
1	Vanja Zivkovic
2	Petar Zivkovic
3	Veljko Zivkovic
4	Nikola Zivkovic



data.names	
0	Andjela
1	Vanja
2	Petar
3	Veljko
4	Nikola

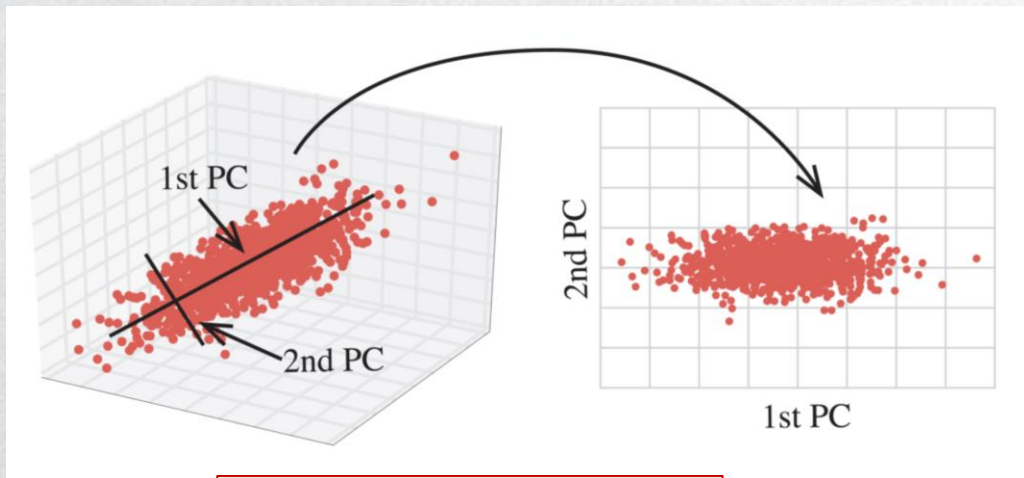
特征工程



特征工程

- 2. Feature Extraction
 - Dimension reduction
 - For example PCA
 - 下图是3维到2维的降维示意图

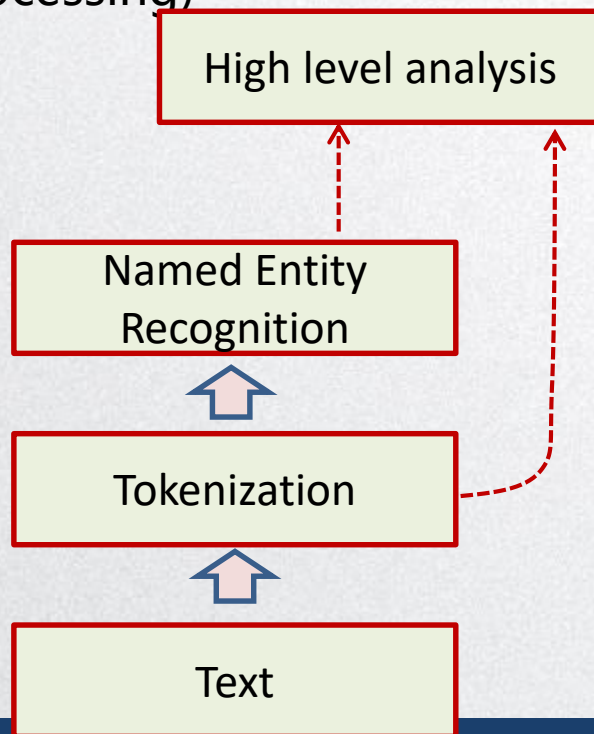
相对于Feature Transformation
Feature Extraction的处理更为深入



PCA的原理这里不展开

特征工程

- 2. Feature Extraction
 - NLP(Natural Language Processing)
 - Tokenization
 - And NER



特征工程

- 2. Feature Extraction
 - Price Time Series Data
 - Ta-Lib
 - Indicators



TA-Lib : Technical Analysis Library

Home

Products
Downloads
Purchase
Support

Function List

Multi-Platform Tools for Market Analysis ...

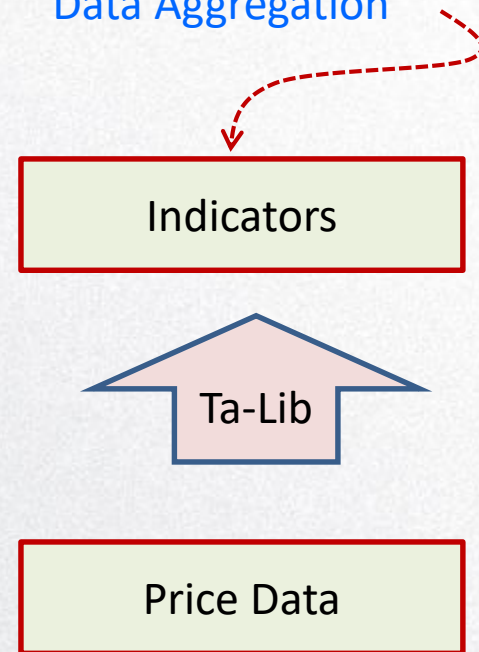
TA-Lib is widely used by trading software developers requiring to perform technical analysis of financial market data.

- Includes 200 indicators such as ADX, MACD, RSI, Stochastic, Bollinger Bands etc... ([more info](#))
- Candlestick pattern recognition
- Open-source API for C/C++, Java, Perl, Python and 100% Managed .NET

Free Open-Source Library

TA-Lib is available under a BSD License allowing it to be integrated in your own open-source or commercial application.

Data Aggregation



特征工程





特征工程

- 3. Feature Selection

- 什么是特征选择
- Feature selection is also known as Variable selection or Attribute selection
- Essentially, it is the process of **selecting the most important/relevant features** of a dataset
 - select those features in your data
 - that contribute most to the prediction variable or output in which you are interested



特征工程

• 3. Feature Selection

- 为什么要进行特征选择
- Often, in a high dimensional dataset, there remain some entirely **irrelevant, insignificant and unimportant features**
 - the contribution of these types of features is often less towards predictive modeling as compared to the critical features
 - They may have zero contribution as well
- Unnecessary resource allocation for these features
 - The machine model takes more time to get trained
- These features act as a noise for which the machine learning model can perform terribly poorly
 - it can make your model very complicated which in turn may lead to overfitting

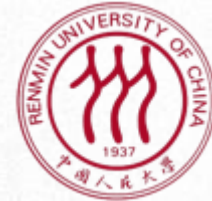
- Feature Selection is the process of selecting out the most significant features from a given dataset
- In many cases, Feature Selection can enhance the performance of a machine learning model as well

特征工程

- 3. Feature Selection

- Benefits/Advantages of Feature Selection/importance of feature selection

- **Reduces Training Time** : enables the machine learning algorithm to **train faster**: Less data means that algorithms train faster
 - **improves the accuracy** of a model if the right subset is chosen: Less misleading data means modeling accuracy improves
 - **reduces Overfitting** : Less redundant data means less opportunity to make decisions based on noise; **Enhanced generalization** by reducing Overfitting
 - **reduces the complexity** of a model and makes it easier to interpret; Simple models are **easier to interpret**
 - Avoid Bad learning behavior in high dimensional spaces



特征工程

- 3. Feature Selection

- Difference **between dimensionality reduction** and **feature selection**
 - Both methods tend to reduce the number of attributes in the dataset
 - but a dimensionality reduction method does so by creating new **combinations** of attributes
 - Some examples of dimensionality reduction methods are Principal Component Analysis(**PCA**), Singular Value Decomposition(**SVD**), Linear Discriminant Analysis(**LDA**)
 - whereas feature selection methods
 - **include and exclude attributes** present in the data without changing them



特征工程

- 3. Feature Selection

- 特征选择的方法 **Types of feature selection**
- 三类方法
 - 1. Filter Method
 - 2. Wrapper Method
 - 3. Embedded Method
- 数值型特征和类别型特征的Selection办法是不一样的

<https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b>

特征工程

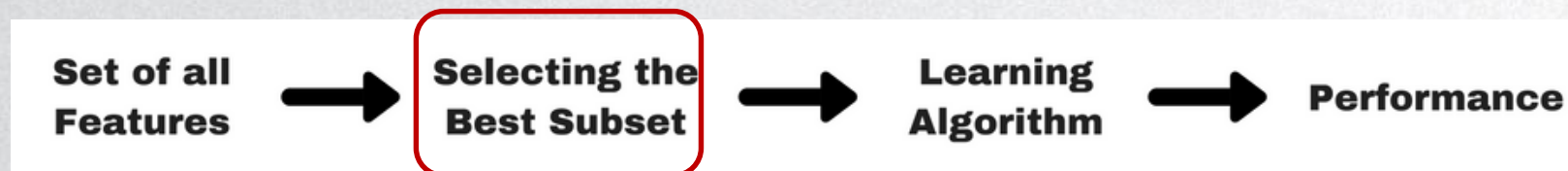


特征工程

- 3. Feature Selection

- 3.1 Filter methods

- Filter method relies on the **general uniqueness of the data** to be evaluated and pick feature subset, not including any mining algorithm
- Filter method uses the exact **assessment criterion** which includes **distance, information, dependency, and consistency**
- Filter methods are generally used as a **data preprocessing** step



不依赖机器学习算法，直接依赖数据的特点（某种评价指标），进行特征选择



特征工程

- 3. Feature Selection
- 3.1 Filter methods
 - Filter methods are generally used as a preprocessing step
 - The selection of features is independent of any machine learning algorithms
 - Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable

特征工程

• 3. Feature Selection

• 3.1 Filter methods

- We can refer to the following table for defining correlation coefficients for different types of data
- Some examples of filter methods include the Chi-squared test, information gain, and correlation coefficients

Pearson相关系数

线性判别分析

Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

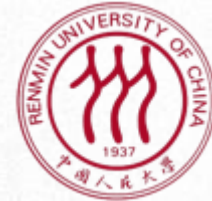
方差分析

卡方测试



特征工程

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.1 Remove constant features
 - Constant features are those that show the same value, just one value, for all the observations of the dataset
 - This is, the same value for all the rows of the dataset
 - These features provide no information that allows a machine learning model to discriminate or predict a target
 - **We can use variance threshold from scikit learn**
 - Variance threshold from scikit learn is a simple baseline approach to feature selection
 - It removes all features which variance doesn't meet some threshold
 - By default, it **removes all zero-variance features**, i.e., features that have the same value in all samples



特征工程

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.2 Remove quasi-constant features
 - Quasi-constant features are those that show the **same value for the great majority of the observations of the dataset**. In general, these features **provide little if any information** that allows a machine learning model to discriminate or predict a target
 - But there can be exceptions, So we should be careful when removing these type of features
 - Identifying and removing quasi-constant features, is an easy first step towards feature selection and more easily interpretable machine learning models
 - To identify quasi-constant features, we can once again use the **Variance Threshold function** from scikit learn
 - Variance threshold from scikit learn is a simple baseline approach to feature selection
 - It removes all features which variance doesn't meet some threshold

特征工程

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.3 Information Gain
 - Information gain calculates the reduction in entropy from the transformation of a dataset
 - It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable
 - Information gain or mutual information measures how much information the presence/absence of a feature contributes to making the correct prediction on the target

信息增益细节请参考



0309-特征工程+ (Information Gain实例) .pptx

特征工程

- 3. Feature Selection

- 3.1 Filter methods

- 3.1.3 Information Gain + `mutual_info_classif` Scikit Learn implementation
 - It estimates mutual information for a **discrete target variable**
 - Mutual information (MI) between two random variables is a non-negative value, which measures the dependency between the variables
 - It is equal to zero if and only if two random variables are independent, and **higher values mean higher dependency**
 - This function relies on nonparametric methods based on entropy estimation from k-nearest neighbors distances
 - It can be used for univariate features selection

特征工程

- 3. Feature Selection

- 3.1 Filter methods

Scikit Learn implementation

- 3.1.3 Information Gain + `mutual_info_regression`

- Estimate mutual information for a **continuous target variable**
- Mutual information (MI) between two random variables is a non-negative value, which measures the dependency between the variables
 - It is equal to zero if and only if two random variables are independent, and **higher values mean higher dependency**
 - The function relies on nonparametric methods based on entropy estimation from k-nearest neighbors distances
 - It can be used for univariate features selection



特征工程

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.4 Chi-square Test
 - The Chi-square test is used for categorical features in a dataset
 - Categorical属性→Categorical 目标变量
 - We calculate Chi-square between each feature and the target
 - and select the desired number of features with the best Chi-square scores

卡方检验细节请参考



0309-特征工程+（卡方检验实例）.pptx

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.4 Chi-square Test - Fisher Score

Scikit Learn
chi-square implementation

- It is the chi-square implementation in scikit-learn
 - It computes chi-squared stats between each **non-negative feature and class**
- This score should be used to evaluate **categorical variables in a classification task**
 - It compares the observed distribution of the different classes of target Y among the different categories of the feature
 - against the expected distribution of the target classes, regardless of the feature categories

- We can perform a **chi-square** test to the samples to retrieve only the two best features from iris dataset



特征工程

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.5 Mean Absolute Difference (MAD)
 - The mean absolute difference (MAD) computes the absolute difference from the mean value
 - The main difference between the variance and MAD measures is the absence of the square in the latter
 - The MAD, like the variance in that higher the MAD, higher the discriminatory power



特征工程

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.6 Variance Threshold
 - The variance threshold is a simple baseline approach to feature selection
 - It removes all features which **variance doesn't meet some threshold**
 - By default, it removes all zero-variance features, i.e., features that have the same value in all samples
 - We assume that features with a higher variance may contain more useful information, but note that we are not taking the relationship between feature variables and target variables into account, which is one of the drawbacks of filter methods



特征工程

方差分析(Analysis of Variance)

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.7 ANOVA F-value For Feature Selection
 - Statistical tests can be used to select those features that have the strongest relationship with the output variable
 - For example the ANOVA F-value method is appropriate for **numerical inputs and categorical data**
 - Compute the ANOVA F-value for the provided sample
 - If the features are categorical, we will calculate a chi-square statistic between each feature and the target vector
 - However, if the **features are quantitative**, we will compute the ANOVA F-value between each feature and the target vector
 - The F-value scores examine if, when we group the numerical feature by the target vector, the means for each group are significantly different

方差分析细节请参考



0309-特征工程+ (ANOVA方差分析实例).pptx

特征工程

Visualize Correlation-Matrix with Heat map

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.8 Correlation Coefficient以及共线性
 - Correlation is a measure of the linear relationship of 2 or more variables. Through correlation, we can predict one variable from the other
 - Good variables are highly correlated with the target
 - Correlated predictor variables provide redundant information
 - Variables should be correlated with the target but uncorrelated among themselves

"Good feature subsets contain features highly correlated with the target, yet uncorrelated to each other"

特征工程

Visualize Correlation-Matrix with Heat map

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.8 Correlation Coefficient以及共线性
 - We can use the Pearson Correlation
 - Using Pearson correlation returned coefficient values will vary between -1 and 1
 - A value closer to 0 implies weaker correlation (exact 0 implying no correlation)
 - A value closer to 1 implies stronger positive correlation
 - A value closer to -1 implies stronger negative correlation

可以保留相关系数大于等于0.5（阈值）的特征

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

特征工程

Visualize Correlation-Matrix with Heat map

- 3. Feature Selection
- 3.1 Filter methods
- 3.1.8 Correlation Coefficient以及共线性
 - The logic behind using correlation for feature selection is that the good variables are highly correlated with the target
 - Furthermore, variables should be correlated with the target but should be uncorrelated among themselves (变量之间不应过多相关)
 - We should check if the selected variables are highly correlated with each other
 - This phenomenon is known as multi-collinearity
 - If they are, we would then need to keep just one of the correlated ones and drop the others



特征工程

- 3. Feature Selection
- 3.1 Filter methods
 - In scikit learn
 - **SelectKBest**(scikit learn) select features according to the k highest scores
 - **SelectPercentile**(scikit learn) Select features according to a percentile of the highest scores
 - SelectKBest and SelectPercentile take as input a scoring function that returns univariate scores
 - For regression tasks: f_regression, mutual_info_regression
 - For classification tasks: chi2, f_classif, mutual_info_classif
 - The methods based on F-test estimate the degree of linear dependency between two random variables. They assume a linear relationship between the feature and the target
 - On the other hand, mutual information methods can capture any kind of statistical dependency, but being nonparametric, they require more samples for accurate estimation



特征工程

- 3. Feature Selection
- 3.1 Filter methods: 小结
 - The characteristics of these methods are as follows
 - These methods rely on the characteristics of the data (feature characteristics)
 - They do not use machine learning algorithms
 - These are model agnostic
 - They tend to be less computationally expensive
 - They usually give lower prediction performance than wrapper methods
 - They are very well suited for a quick screen and removal of irrelevant features

agnostic

adj.

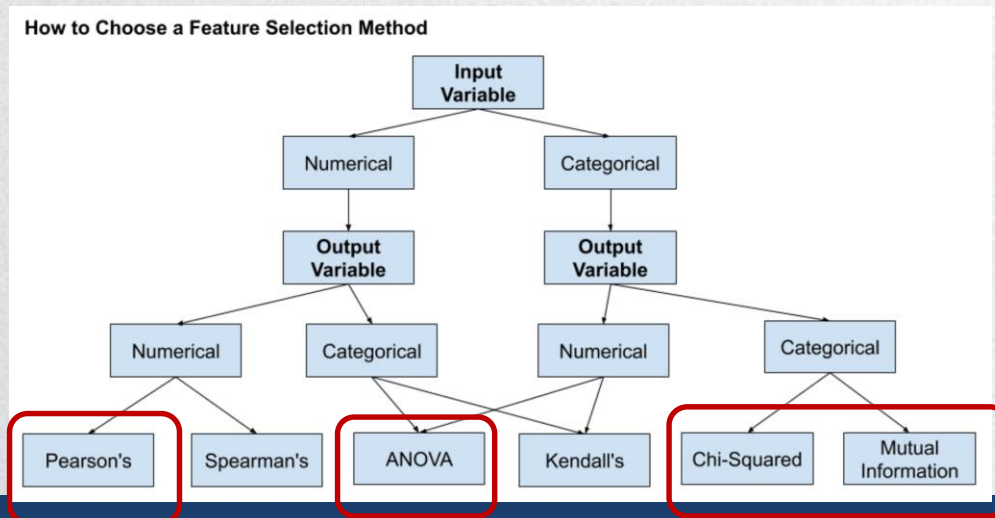
不可知论 (者) 的;

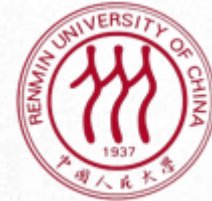
特征工程



特征工程

- 3. Feature Selection
- 3.1 Filter methods
 - **How to choose the right feature selection method**
 - We can see that there are lot of feature selection techniques available
 - The following graphic will serve as a guide on how to choose a feature selection method





特征工程

- 3. Feature Selection

- 3.1 Filter methods

- Numerical Input, Numerical Output

- This is a regression predictive modeling problem with numerical input variables
 - The most common techniques are to use a correlation coefficient, such as Pearson' s for a linear correlation, or rank-based methods for a nonlinear correlation
 - The tests employed are as follows
 - Pearson' s correlation coefficient (linear)
 - Spearman' s rank coefficient (nonlinear)



特征工程

- 3. Feature Selection
- 3.1 Filter methods
 - Numerical Input, Categorical Output
 - This is a classification predictive modeling problem with numerical input variables
 - This might be the most common example of a classification problem
 - Again, the most common techniques are correlation based, although in this case, they must take the categorical target into account
 - We can employ the following tests as follows
 - ANOVA correlation coefficient (linear)
 - Kendall's rank coefficient (nonlinear)
 - Kendall does assume that the categorical variable is ordinal



特征工程

- 3. Feature Selection
- 3.1 Filter methods
 - Categorical Input, Numerical Output
 - This is a regression predictive modeling problem with categorical input variables
 - This is a strange example of a regression problem (e.g. we will not encounter it often)
 - We can use the same “Numerical Input, Categorical Output” methods (described previously), but in reverse



特征工程

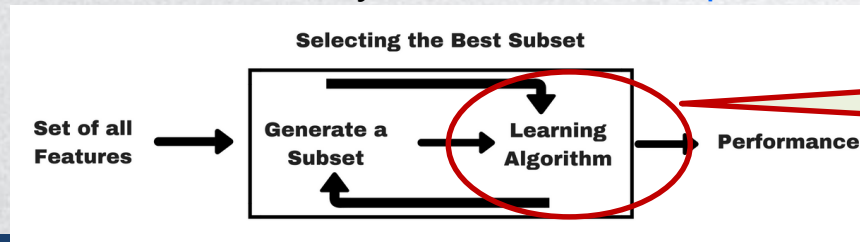
- 3. Feature Selection
- 3.1 Filter methods
 - Categorical Input, Categorical Output
 - This is a classification predictive modeling problem with categorical input variables
 - The most common correlation measure for categorical data is the **chi-squared test (contingency tables)**
 - We can also use **mutual information /information gain** from the field of information theory
 - In fact, mutual information is a powerful method that may prove useful for both categorical and numerical data, e.g. it is agnostic to the data types

特征工程



特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
 - a wrapper method needs one machine learning algorithm and uses its performance as evaluation criteria to evaluate features(independent variables)
 - This method searches for a feature sub set which is best-suited for the machine learning algorithm and aims to improve the mining performance
 - To evaluate the features, the predictive accuracy used for classification tasks and goodness of cluster is evaluated using clustering
 - The problem is essentially reduced to a search problem, some methods follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion; These methods are usually computationally very expensive
 - The wrapper methods usually result in better predictive accuracy than filter methods



使用learning algorithm



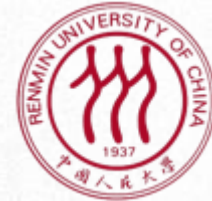
特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
 - Wrapper methods consists of the following techniques
 - Forward Selection
 - Backward Elimination
 - Exhaustive Feature Selection
 - Recursive Feature Elimination
 - Recursive Feature Elimination with Cross-Validation



特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
- 3.2.1 Forward Selection
 - Forward selection is an iterative method in which we start with having no feature in the model
 - In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model
 - The procedure starts with an empty set of features. The best of the original features is determined and added to the reduced set
 - At each subsequent iteration, the best of the remaining original attributes is added to the set
 - Step forward feature selection starts by evaluating all features individually and selects the one that generates the best performing algorithm, according to a pre-set evaluation criteria
 - In the second step, it evaluates all possible combinations of the selected feature and a second feature, and selects the pair that produce the best performing algorithm based on the same pre-set criteria



特征工程

- 3. Feature Selection
- 3.2 Wrapper methods

mlxtend python library has implementation

- 3.2.1 Forward Selection

- The pre-set criteria can be the **ROC/AUC for classification and the R squared for regression for example**
- This selection procedure is called greedy, because it evaluates all possible single, double, triple and so on feature combinations. Therefore, it is quite computationally expensive, and sometimes, if feature space is big, even unfeasible

- ROC(Receiver Operating Characteristic)曲线
- AUC(Area Under the Curve)面积
- 评价分类模型的指标

特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
- 3.2.2 Backward Elimination

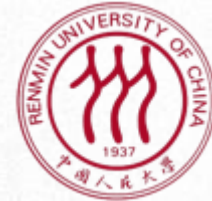
mlxtend python library has implementation

- In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model
 - The procedure starts with the full set of attributes
 - At each step, it removes the worst attribute remaining in the set
 - We repeat this until no improvement is observed on removal of features

特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
- 3.2.3 Combination of forward selection and backward elimination
 - The stepwise forward selection and backward elimination methods can be **combined**
 - so that, **at each step**, the procedure **selects the best attribute** from among the remaining attributes and **removes the worst**

mlxtend python library has implementation



特征工程

- 3. Feature Selection

- 3.2 Wrapper methods

mlxtend python library has implementation

- 3.2.4 Exhaustive Feature Selection

- It is a brute-force evaluation of each feature subset. In an exhaustive feature selection the best subset of features is selected, **over all possible feature subsets**, by optimizing a specified performance metric for a certain machine learning algorithm
 - For example, if the classifier is a logistic regression and the dataset consists of 4 features, the algorithm will evaluate all $15(2^N - 1)$ feature combinations as follows:
 - all possible combinations of 1 feature/ all possible combinations of 2 features/ all possible combinations of 3 features/ all the 4 features
 - and select the one that results in the best performance (e.g., classification accuracy) of the logistic regression classifier
- This is **greedy algorithm** as it **evaluates all possible feature combinations**
 - This is the most robust feature selection method covered so far
 - It is quite computationally expensive, and sometimes, if feature space is big, even unfeasible



特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
- 3.2.5 Recursive Feature elimination
 - It is a greedy optimization algorithm which aims to find the best performing feature subset
 - The Recursive Feature Elimination (or RFE) works by **recursively removing attributes** and **building a model on those attributes that remain**
 - It uses the **model accuracy** to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute
 - RFE selects features by considering a smaller and smaller set of features(regressors)
 - Recursive feature elimination technique eliminates n features from a model by **fitting the model multiple times and at each step, removing the weakest features**



特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
- 3.2.5 Recursive Feature elimination
 - The starting point is the original set of features
 - Less important features are recursively pruned from the initial set
 - The procedure is repeated until a desired set of features remain
 - number can either be a priori specified, or can be found using cross validation

特征工程

- 3. Feature Selection
 - 3.2 Wrapper methods
 - 3.2.5 Recursive Feature elimination
 - The RFE method takes the **model** to be used and **the number of required features as input**
 - It then gives the **ranking** of all the variables, 1 being most important
 - It also gives its **support**, True being relevant feature and False being irrelevant feature
- First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a **coef_attribute**(e.g., the **coefficients of a linear model**) or through a **feature_importances_** attribute
 - Then, the **least important features** are pruned from the current set of features
 - That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached
 - It then ranks the features based on the order of their elimination



特征工程

- 3. Feature Selection
- 3.2 Wrapper methods
- 3.2.5 Recursive Feature elimination with Cross-Validated (RFECV)
 - In scikit learn, RFE offers a variant – RFECV – designed to optimally find the best subset of regressors
 - Recursive Feature Elimination feature selection technique selects the best subset of features for the estimator by removing 0 to N features iteratively using recursive feature elimination
 - Then it selects the best subset based on **cross-validation score**
 - the accuracy or ROC/AUC of the model

Recursive Feature elimination 和 cross-validation 结合

特征工程



特征工程

- **Difference between filter and wrapper methods**

- It might get confusing to differentiate between filter methods and wrapper methods in terms of their functionalities
 - Filter methods **do not** incorporate a **machine learning model** in order to determine if a feature is good or bad
 - whereas wrapper methods use a machine learning model and train it the feature to decide if it is essential or not
 - Filter methods are much **faster** compared to wrapper methods as they do not involve training the models
 - On the other hand, wrapper methods are computationally costly, and in the case of massive datasets, wrapper methods are not the most effective feature selection method to consider. When dealing with high-dimensional data, it is computationally cheaper to use filter methods

特征工程

- **Difference between filter and wrapper methods**

- (continue)

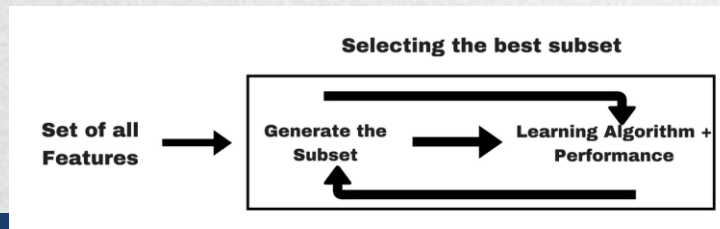
- Filter methods **may fail** to find the best subset of features in situations when there is not enough data to model the statistical correlation of the features
 - but wrapper methods can always provide the best subset of features because of their exhaustive nature
 - Using features from wrapper methods in your final machine learning model **can lead to overfitting** as wrapper methods already train machine learning models with the features and it affects the true power of learning
 - But the features from filter methods will not lead to overfitting in most of the cases

特征工程



特征工程

- 3. Feature Selection
- 3.3 Embedded methods
 - In Embedded Methods, the feature selection algorithm is integrated as part of the learning algorithm.
 - **Regularization** methods are the most commonly used embedded methods which penalize a feature given a coefficient threshold
 - Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients)
 - Some of the most popular examples of these methods are **LASSO, RIDGE and Elastic Net regression** which have inbuilt penalization functions to reduce overfitting





特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.1 Regularization methods—LASSO, Ridge Regression, Elastic Net
 - LASSO Regression (L1) Based Feature Selection
 - Lasso regression performs L1 regularization which adds penalty equivalent to absolute value of the magnitude of coefficients
 - Regularization consists in adding a penalty to the different parameters of the machine learning model to reduce the freedom of the model and in other words to avoid overfitting
 - In linear model regularization, the penalty is applied over the coefficients that multiply each of the predictors
 - From the different types of regularization, Lasso or L1 has the property that is able to shrink some of the coefficients to zero
 - If the feature is irrelevant, lasso penalizes its coefficient and make it 0
 - Therefore, that feature(coefficient = 0) can be removed from the model

请参考多元线性回归的“正则化”



特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.1 Regularization methods—LASSO, Ridge Regression, Elastic Net
 - LASSO Regression (L1) Based Feature Selection
 - Lasso regularization helps to remove non-important features from the dataset
 - increasing the penalization will result in increase the number of features removed
 - Therefore, we need to keep an eye and monitor that we don't set a penalty too high so that to remove even important features, or too low and then not remove non-important features.
 - If the penalty is too high and important features are removed, we will notice a drop in the performance of the algorithm and then realize that we need to decrease the regularization

请参考多元线性回归的“正则化”



特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.1 Regularization methods——LASSO, Ridge Regression , Elastic Net

此处不展开



特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.1 Regularization methods——LASSO, Ridge Regression, Elastic Net

此处不展开



特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.2 Random Forest Importance(Tree-based Feature Selection)
 - Decision trees or other tree-based models contain a variable importance output that can be used to decide, which feature to select for inclusion
 - Features that are closer to the root of the tree are more important than those at end splits, which are not as relevant

特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.2 Random Forest Importance(Tree-based Feature Selection)

- Random forests are one the most popular machine learning algorithms. They are so successful because they provide in general a good predictive performance, low overfitting and easy interpretability
 - This interpretability is given by the fact that it is straightforward to derive the importance of each variable on the tree decision
 - In other words, it is easy to compute how much each variable is contributing to the decision

特征工程

- 3. Feature Selection
- 3.3 Embedded methods
- 3.3.2 Random Forest Importance(Tree-based Feature Selection)
 - Random forests consist of 4-12 hundred decision trees, each of them built over a random extraction of the observations from the dataset and a random extraction of the features. **Not every tree sees all the features** or all the observations, and this guarantees that the trees are de-correlated and therefore less prone to over-fitting
 - Each tree is also a sequence of yes-no questions based on a single or combination of features. At each node (this is at each question), the feature divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket
 - Therefore, the **importance of each feature** is derived **by how "pure" each of the buckets is**



特征工程

- 3. Feature Selection
 - 3.3 Embedded methods
 - 3.3.2 Random Forest Importance(Tree-based Feature Selection)
 - For classification, the measure of impurity is either the **Gini impurity** or the **information gain**. For regression the measure of impurity is variance
 - Therefore, when training a tree, it is possible to compute how much each feature decreases the impurity. **The more a feature decreases the impurity, the more important the feature is**
 - In random forests, the impurity decrease from each feature can be **averaged** across trees to determine the final importance of the variable
-
- To give a better intuition, features that are selected at the **top of the trees** are in general more important than features that are selected at the **end nodes of the trees**, as generally the top splits lead to bigger information gains
 - **by pruning trees below a particular node, we can create a subset of the most important features**

R
U
C

Q
X
P

特征工程





特征工程

- Choose the right feature selection methods
 - Filter method vs. Wrapper method vs. Embedded method
 - Filter method is less accurate
 - It is great while doing EDA, it can also be used for checking multi co-linearity in data
 - Wrapper and Embedded methods give more accurate results but as they are computationally expensive
 - these method are suited when you have less features (~20)

特征工程





特征工程

- 特征工程实践

名称	类型	大小	修改日期
 00_feature_engineering_penguins.ipynb	IPYNB 文件	128 KB	2021/12/28 19:31

(1)

1. Imputation – missing values
2. Categorical Encoding
 - 2.1 Label Encoding
 - 2.2 One-Hot Encoding
 - 2.3 Count Encoding
 - 2.4 Target Encoding
 - 2.5 Leave One Out Target Encoding
3. Handling Outliers
4. Binning

5. Scaling




- 5.1 Standard Scaling
 - 5.2 Min-Max Scaling (Normalization)
 - 5.3 Quantile Transformation
6. Log Transform
 7. Feature Selection using SelectKBest(f_classif)
 8. Feature Grouping
 9. Feature Split

U
C
Q
X
P



特征工程

- 特征工程实践

名称	类型	大小	修改日期
 01_feature_selection_boston_house_price.ipynb	IPYNB 文件	437 KB	2021/12/21 17:22
 01_feature_selection_boston_house_price_Filter_Wrapper_Embedded.ipynb	IPYNB 文件	150 KB	2021/12/21 17:11
 01_feature_significance_boston_house_price_LinearReg_Lasso_Ridge.ipynb	IPYNB 文件	118 KB	2021/12/25 14:20

(1)

1. Regression forward selection
2. Regression backward selection
3. Regression stepwise selection
4. REF \leftarrow Linear Regression
- 4.1 REFCV \leftarrow Linear Regression
5. SelectKBest(score_func=f_regression)
6. Correlation
7. Lasso (L1) Based Feature Selection
8. Tree-based Feature Selection

(2)

1. Filter method \leftarrow correlation
2. Wrapper Method \leftarrow Backward Elimination
3. Wrapper Method \leftarrow REF (Linear Regression)
4. Embedded method \leftarrow Lasso


(3)

1. Linear regression(importances =lm.coef_)
- 2.Lasso regression (importances =lm.coef_)
3. Ridge regression (importances =lm.coef_)



特征工程

- 特征工程实践

名称	类型	大小	修改日期
 09_feature_selection_pima-indians-diabetes (1) .ipynb	IPYNB 文件	40 KB	2022/2/12 13:38
 09_feature_selection_pima-indians-diabetes (2) .ipynb	IPYNB 文件	20 KB	2022/2/12 13:40
 09_feature_selection_pima-indians-diabetes (3) .ipynb	IPYNB 文件	221 KB	2022/2/12 13:45

(1)

1. SelectKBest \leftarrow score_func=f_classif
2. RFE \leftarrow Logistic Regression
3. PCA \leftarrow n_components=3
4. ExtraTreesClassifier \leftarrow feature_importances_

(2)

1. SelectKBest \leftarrow score_func=chi2
2. REF \leftarrow Logistic Regression
3. Regularization \leftarrow Ridge

(3)

1. Filter method \leftarrow information gain(mutual_info_classif)
2. Filter method \leftarrow Chi-square Test SelectKBest(chi2)
3. Filter Method \leftarrow Correlation Coefficient
4. Filter method \leftarrow Variance Threshold
5. Filter method \leftarrow Mean Absolute Difference (MAD)
6. Wrapper Methods \leftarrow Forward Feature Selection
7. Wrapper Methods \leftarrow Backward Feature Selection
8. Wrapper Methods \leftarrow Exhaustive Feature Selection
9. Wrapper Methods \leftarrow Recursive Feature Elimination
10. Embedded Methods \leftarrow LASSO Regularization (L1)
11. Embedded Methods \leftarrow Random Forest Importance