Train In Data

Selecting features with Mutual Information

Mutual Information

Mutual information is a measure of the mutual association of 2 variables.

How do we use it to select features?

Mutual Information

Mutual information is a measure of the mutual association of 2 variables.

How do we use it to select features?

We rank features based on their MI and then select the top ranking features.

Mutual Information: Scikit-learn

Feat 1	Feat 2	Feat 3	Target
2	Blue	20	0
3	Red	22	1
3	Green	25	0
7	Green	46	0
10	Red	30	1
5	Blue	36	0
4	Red	37	1

- **Step 1**: determine mutual information between each feature and the target

Mutual Information: Scikit-learn

Feat 1	Feat 2	Feat 3	Target
2	Blue	20	0
3	Red	22	1
3	Green	25	0
7	Green	46	0
10	Red	30	1
5	Blue	36	0
4	Red	37	1

- **Step 1**: determine mutual information between each feature and the target

0.31	**0.45**	**0.20**

Mutual information

Mutual Information: Scikit-learn

Feat 1	Feat 2	Feat 3	Target
2	Blue	20	0
3	Red	22	1
3	Green	25	0
7	Green	46	0
10	Red	30	1
5	Blue	36	0
4	Red	37	1

0.31	0.45	0.20

Mutual information

- **Step 2**: rank the features based on the mutual information
  - Feat 2
  - Feat 1
  - Feat 3

Mutual Information: Scikit-learn

Feat 1	Feat 2	Feat 3	Target
2	Blue	20	0
3	Red	22	1
3	Green	25	0
7	Green	46	0
10	Red	30	1
5	Blue	36	0
4	Red	37	1

0.31	0.45	0.20

Mutual information

- **Step 3**: select top ranking feature
  - **Feat 2**
  - Feat 1
  - Feat 3

Mutual Information: Scikit-learn

Step 3

- `SelectKBest`
  - Selects top k highest ranking features, e.g., top 10 features.
- `SelectPercentile`
  - Selects features in the top percentile, i.e., top 10th percentile.
- Get creative
  - Select features with MI > mean(MI)

Mutual Information: Scikit-learn

Step 1

- `mutual_info_classif` (discrete target)
- `mutual_info_regression` (continuous target)

MI - sklearn

Important for the correct calculation of MI

Larger k, smaller "bins" or smaller neighbourhood

KNN has an element of randomness, so set the seed!

mutual_info_regression

```
sklearn.feature_selection.mutual_info_regression(X, y, *, discrete_features='auto',
n_neighbors=3, copy=True, random_state=None, n_jobs=None)
```
[source]

Estimate mutual information for a continuous target variable.

Mutual information (MI) [1] 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 as described in [2] and [3]. Both methods are based on the idea originally proposed in [4].

It can be used for univariate features selection, read more in the User Guide.

Parameters:

X : *array-like or sparse matrix, shape (n_samples, n_features)*

Feature matrix.

y : *array-like of shape (n_samples,)*

Target vector.

discrete_features : *{'auto', bool, array-like), default='auto'*

If bool, then determines whether to consider all features discrete or continuous. If array, then it should be either a boolean mask with shape (n_features,) or array with indices of discrete features. If 'auto', it is assigned to False for dense `X` and to True for sparse `X`.

n_neighbors : *int, default=3*

Number of neighbors to use for MI estimation for continuous variables, see [2] and [3]. Higher values reduce variance of the estimation, but could introduce a bias.

copy : *bool, default=True*

Whether to make a copy of the given data. If set to False, the initial data will be overwritten.

random_state : *int, RandomState instance or None, default=None*

Determines random number generation for adding small noise to continuous variables in order to remove repeated values. Pass an int for reproducible results across multiple function calls. See Glossary.

n_jobs : *int, default=None*

The number of jobs to use for computing the mutual information. The parallelization is done on the columns of `X`.

`None` means 1 unless in a `joblib.parallel_backend` context. `-1` means using all processors. See Glossary for more details.

Train In Data

THANK YOU

www.trainindata.com