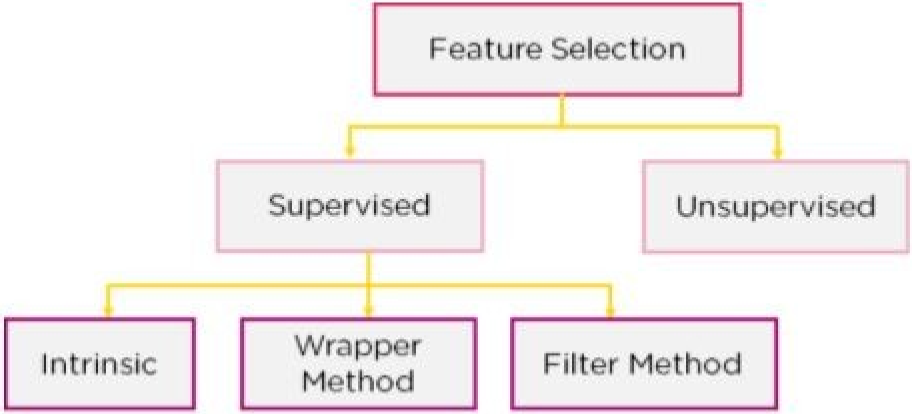
Feature Selection

Allgemeine Übersicht über Feature Selection
Verfahrensarten und Algorithmen

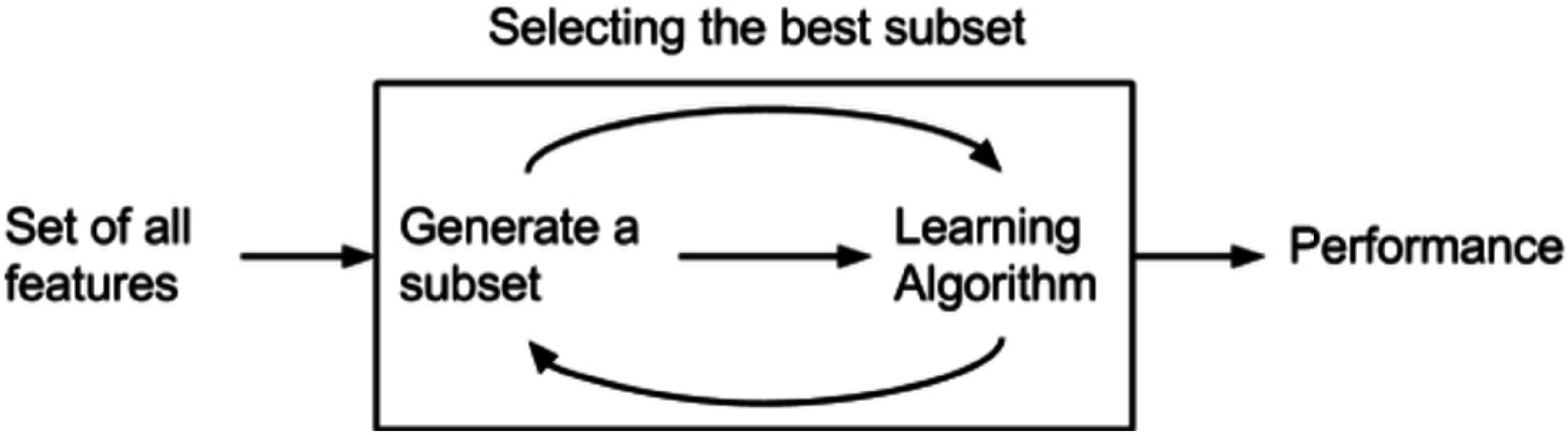
Anfang Recherche Feature Selection mit NAS


Feature Selection
Supervised
Unsupervised
Intrinsic
Wrapper Method
Filter Method

Feature Selection: Select a subset of input features from the dataset.

- Unsupervised: Do not use the target variable (e.g. remove redundant variables).
  - Correlation
- Supervised: Use the target variable (e.g. remove irrelevant variables).
  - Wrapper: Search for well-performing subsets of features.
    - Recursive Feature Elimination (RFE)
  - Filter: Select subsets of features based on their relationship with the target.
    - Statistical Methods
    - Feature Importance Methods
- Intrinsic/Embedded: Algorithms that perform automatic feature selection during training.
  - Decision Trees
  - Regularizers (Lasso)
- Dimensionality Reduction: Project input data into a lower-dimensional feature space

Wrapper


Selecting the best subset
Set of all features
Generate a subset
Learning Algorithm
Performance

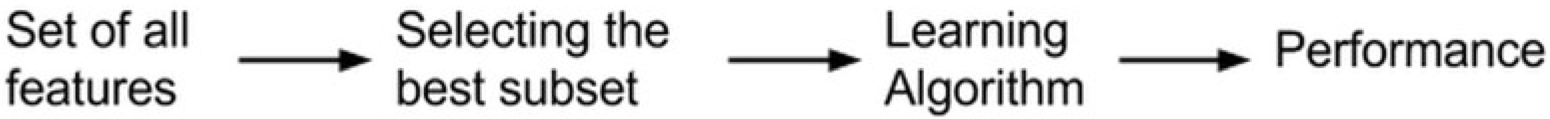
Wrapper RFE

- *The goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef_ attribute or through a feature_importances_ attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.*
- As you would have guessed, we could use any estimator with the method. In this case, we use `LogisticRegression` , and the RFE observes the `coef_` attribute of the `LogisticRegression` object

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
rfe_selector = RFE(estimator=LogisticRegression(), n_features_to_select=num_feats, step=10, verbose
rfe_selector.fit(X_norm, y)
rfe_support = rfe_selector.get_support()
rfe_feature = X.loc[:,rfe_support].columns.tolist()
print(str(len(rfe_feature)), 'selected features')
```

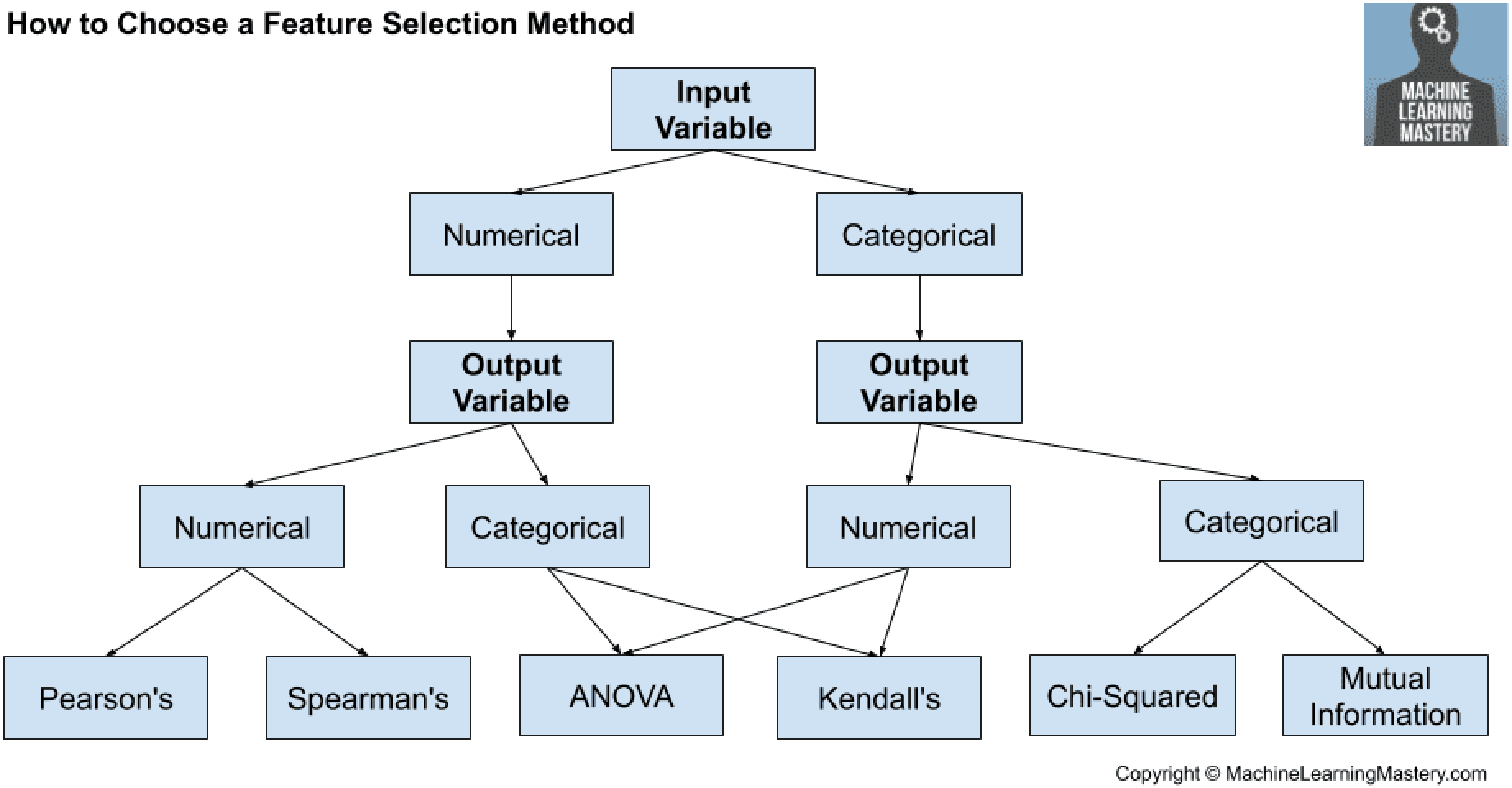
rfe.py hosted with ❤ by GitHub view raw

Filter



Set of all features
Selecting the best subset
Learning Algorithm
Performance

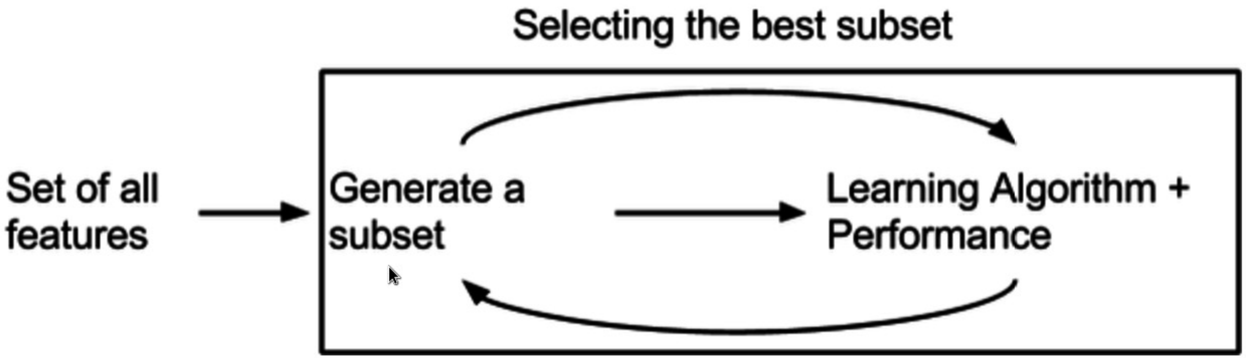
Filter


How to Choose a Feature Selection Method
Input Variable
Numerical
Categorical
Output Variable
Output Variable
Numerical
Categorical
Numerical
Categorical
Pearson's
Spearman's
ANOVA
Kendall's
Chi-Squared
Mutual Information
MACHINE LEARNING MASTERY
Copyright © MachineLearningMastery.com

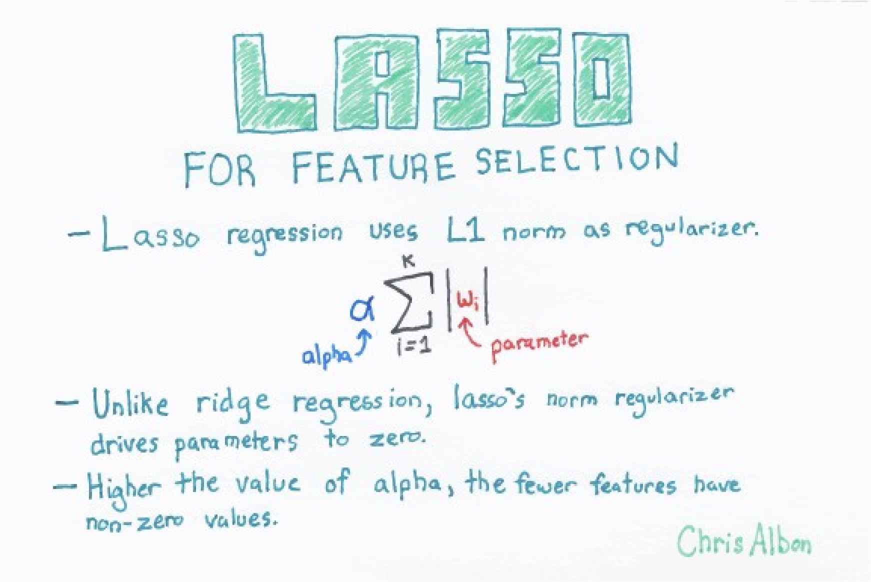
Filter: Numerical Input, Numerical Output

- Pearson’s correlation coefficient (linear)
  - sklearn.feature_selection.f_regression
- Spearman’s rank coefficient (nonlinear)
  - scipy.stats.spearmanr

Intrinsic/Embedded



Intrinsic/embedded




```python
from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import LogisticRegression

embeded_lr_selector = SelectFromModel(LogisticRegression(penalty="l1"), max_features=num_feats)
embeded_lr_selector.fit(X_norm, y)

embeded_lr_support = embeded_lr_selector.get_support()
embeded_lr_feature = X.loc[:,embeded_lr_support].columns.tolist()
print(str(len(embeded_lr_feature)), 'selected features')
```

lasso.py hosted with ❤ by GitHub view raw

Intrinsic/embedded

- We can use RandomForest to select features based on feature importance.
- We calculate feature importance using node impurities in each decision tree. In Random forest, the final feature importance is the average of all decision tree feature importance.

AutoEncoders

- Can we use AutoEncoders for Feature Selection (not Dimensionality Reduction)?

Feature Selection NAS (auto pytorch)

- AutoPyTorch Paper: ConfigurationSpace
- PyTorch Data preprocessing: https://pytorch.org/tutorials/beginner/basics/transforms_tutorial.html

https://arxiv.org/pdf/2006.13799.pdf