Machine Learning

Feature Selection & Engineering

Part Two

Dr. Sherif Saad

Learning Objectives

Introduce the students to feature engineering

Understand the importance of feature engineering.

Understand the problems feature engineering solve.

Hand-on experience with feature engineering.

Outlines

- Introduction to Feature Selection
- Feature Selection Algorithms and Techniques
- Feature Selection with Scikit-Learn

Feature Selection: The What

After features brainstorming and devising, the next task focus on answering the following question: Which features we should use to create the predictive model?

Feature Selection is the process of selecting the most relevant features from the features space that are relevant or significant to the problem you are trying to solve.

Domain knowledge of the problem we are trying to solve is a great asset while we are selecting the most relevant features

Feature Selection: The What

The process of feature selection is kind of a filter that does not construct new feature or aggregate existing features.

Dimensionality reduction on the other hand aggregate and fuse two or more features into one new feature to reduce the number of features we will use to build the predictive model.

Dimensionality reduction is an application of unsupervised learning methods.

It is inaccurate to claim that dimensionality reduction is a feature selection method. [WHY?]

Feature Selection: The Why

Why we create or collect as many features as we can and then only select a subset of these features to build our predictive model?

Feature selection help us in identifying and removing irrelevant features from the data that do not contribute in improving the accuracy of our ML model or could decrease the accuracy of the model.

Irrelevant features are redundant features and noisy features. In general, redundant and noisy features have negative impact on the quality of the prediction model

Feature Selection: The Why

The key benefits we obtain from applying feature selection are:

- Improve the accuracy of the prediction model. Because we eliminate misleading features.
- Reduce overfitting; redundant attribute usually result in overfitting the model to the training data (e.g. KNN and Decision Tree)
- Reduce training time; by reducing the number of features, we reduce the size of the input. This means that ML algorithms will require less time to learn and construct the model

Feature Selection: The How

In general, algorithms and methods for feature selection are categorized into three main categories, namely, filter methods, wrapper methods, and embedded methods.

Filter Methods:

- Assign a score or rank to each feature using some statistical methods.
- Based on the feature rank or score we decide if we will keep or drop the feature.
- These methods usually univariate and consider the feature independently.
- Examples: Information Gain, Chi-square, and Gini-Index

Feature Selection: The How

Wrapper Methods:

- Model the feature selection process and as a search problem.
- Prepare different combinations (subsets) from the features space.
- Evaluate each combination and compare it to the other combinations.
- The search process could be best-first search, greedy, or stochastic such as hill-climbing algorithm or genetic algorithm.
- Example: recursive feature elimination algorithm.

Feature Selection: The How

Embedded Methods:

- Learning the best features to build the model while the model is being created.
- Regularization or penalization algorithm are the most common embedded feature selection methods.
- Embedded methods are very common in regression problems.
- Example of regularization algorithms are Elastic Net, LASSO, and Ride Regression.

Feature Selection: The How

Are decision trees embedded methods or filter methods?

If we are using genetic algorithm for feature selection, what will be the fitness function?

Feature Selection: Recursive Feature Elimination

Is a very simple and powerful algorithm for feature selection.

The idea is very simple, recursively remove a feature and build the model using the remaining features.

Evaluate the model based on the selected subset of features and use the model accuracy to estimate the rank of the rank of the excluded feature.

Recursive Feature Elimination - Scikit-Learn

```
# We use scikit-learn examples (toy) datasets
from sklearn import datasets

# from the feature selection model we import the RFE
from sklearn.feature_selection import RFE

from sklearn.linear_model import LogisticRegression

# load the iris datasets
dataset = datasets.load_iris()

# create a base classifier used to evaluate a subset of attributes
model = LogisticRegression()

# create the RFE model and select 3 attributes
rfe = RFE(model, 3)
rfe = rfe.fit(dataset.data, dataset.target)

# summarize the selection of the attributes
print(dataset.feature_names)
print(rfe.support_)
print(rfe.ranking_)
```

Recursive Feature Elimination - Scikit-Learn

```
Time  Line #  Log Message
1.4s       0  ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
              [False  True  True  True]
1.4s       1  [2 1 1 1]
```

Feature Selection: Ensembles Algorithm

Use ensembles of decision trees to compute the relative importance of each attribute.

Implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

The importance values can be used to inform a feature selection process. The higher the score, the more important the feature

Ensembles Algorithm - Scikit-Learn

```
# We use scikit-learn examples (toy) datasets
from sklearn import datasets

# We load an ensemble model
from sklearn.ensemble import ExtraTreesClassifier

# load the iris datasets
dataset = datasets.load_iris()

# fit an Extra Trees model to the data
model = ExtraTreesClassifier()
model.fit(dataset.data, dataset.target)

# display the relative importance of each attribute
print(dataset.feature_names)
print(model.feature_importances_)
```

Ensembles Algorithm - Scikit-Learn

```
Time  Line #  Log Message
1.3s       0  ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
1.4s       1  [ 0.12679237  0.03972643  0.31200744  0.52147376]
```

Feature Selection Relief Algorithm

It is a feature selection algorithm that is commonly used with binary classification problems. I was proposed in 1992 by Kira and Rendell

It does not depend on heuristics and it has a low-order polynomial time (has a good or accepted run time)

It can tolerate noise to some extend and works well with discrete and continuous features.

It needs large number of samples and observation to avoid getting stuck in local optima. Finally it can not detect redundant features.

Feature Selection Relief Algorithm

Inputs:

- ***N*** number of samples, each samples has ***M*** number of features.
- All the samples have labels (classes)

Preprocessing:

- All the features must be scaled between [0,1]

Output:

- A vector of length ***M*** that represent the weights (features importance)

Feature Selection Relief Algorithm

Operations:

1. It is an iterative process and at each iteration the algorithm randomly select a sample X
2. Then it select two other samples, namely, near-Hit and near-Miss
3. **near-Hit:** the closest sample in the dataset to X that share the same label of X
4. **near-Miss:** the closest sample in the dataset to X that does not share the same label of X

Feature Selection Relief Algorithm

Operations:

5. It update the weights vector (stores the weights of each feature), using the following equation

$$W_i = W_i - (x_i - nearHit_i)^2 + (x_i - nearMiss - i)^2$$

6. The weight of any given feature decreases if it differs from that feature in nearby instances of the same class more than nearby instances of the other class, and increases in the reverse case.

Feature Selection Relief Algorithm

Operations:

7. After *t* iterations, divide each element of the weight vector by *t*. Which give us the relevance vector of the features.
8. Finally the features are selected if their relevance is greater than a predefined threshold.

To which type or category of the feature selection methods the RELIEF algorithm belongs?

Questions