

Feature Selection



Giriş

Sınıflandırma ve regresyonun amacı, sonuç ile ilişkilendirilen featurelarla bir fonksiyon öğrenmektir.

- Genellikle çok fazla feature bulunur.
- Bunların hepsi alakalı olmaz.

Giriş

Feature selection, sonucu doğru tahmin etmek için yararlı olan "optimal" feature belirleme görevidir.

Motivasyon

Accuracy

Alakasız featurelardan kurtulmak, karışıklığı azaltarak daha iyi tahmine dayalı modellerin öğrenilmesine yardımcı olur.

Generalizability

Daha az özelliğe sahip modeller daha düşük karmaşıklığa sahiptir, bu nedenle overfite daha az eğimli olur.

Motivasyon

Interpretability

Az feature seçmek, etiket ile arasındaki ilişkinin mekaniğini anlamaya yardımcı olur.

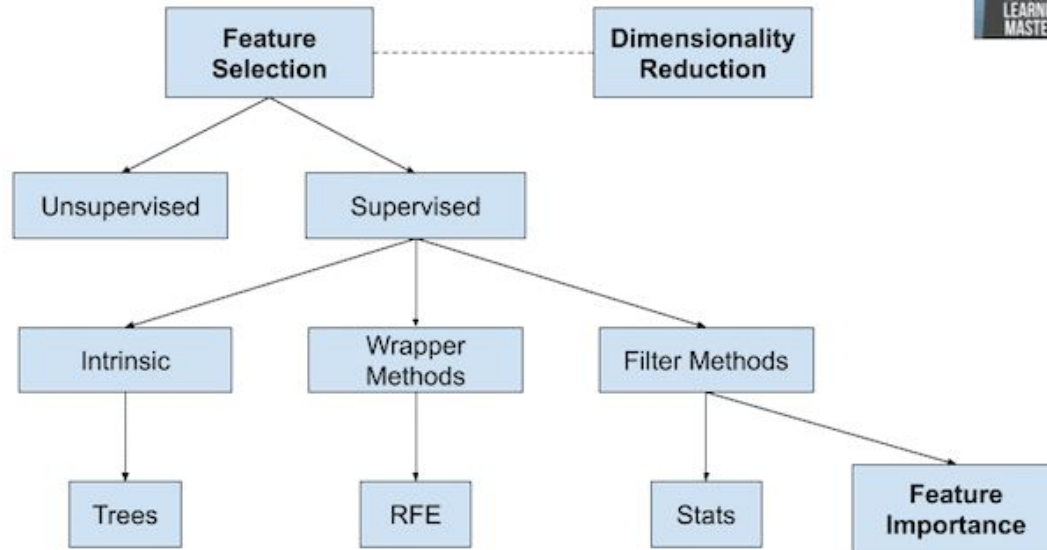
Efficiency

Az sayıda feature kullanılırsa, zamandan ve alandan kazanç sağlanır.

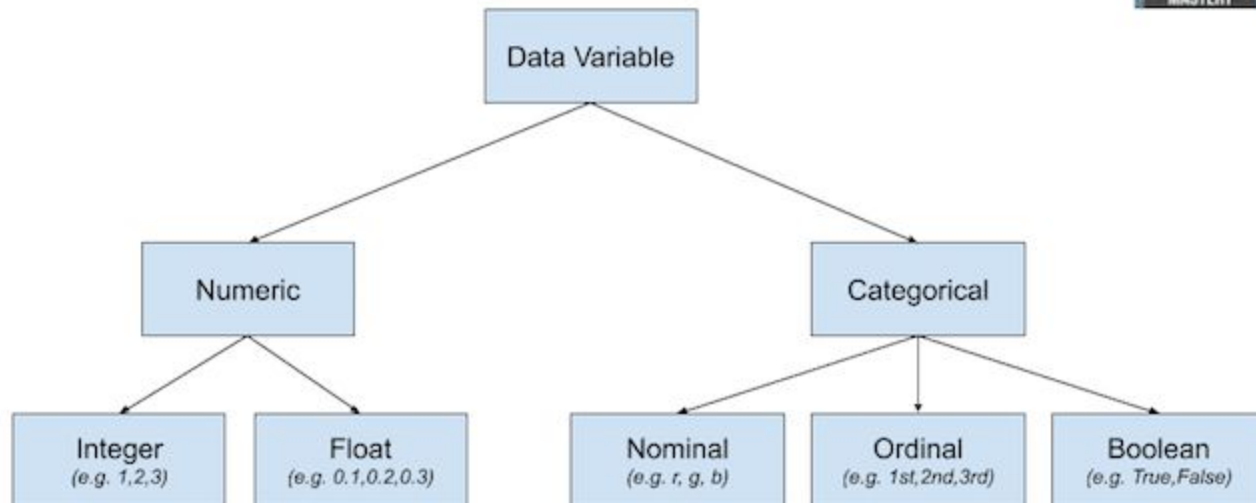
Yöntem

1. **Feature seçimi ayrı görev olarak ele al.**
 - Filtering-based feature selection
 - Wrapper-based feature selection
2. **Feature seçimi bir model öğrenme görevi olarak ele al.**
 - Regularization
3. **Feature seçme, feature kombinasyonlarını etkin bir şekilde temsil eden yeni featurelar oluştur.**
 - Dimensionality reduction

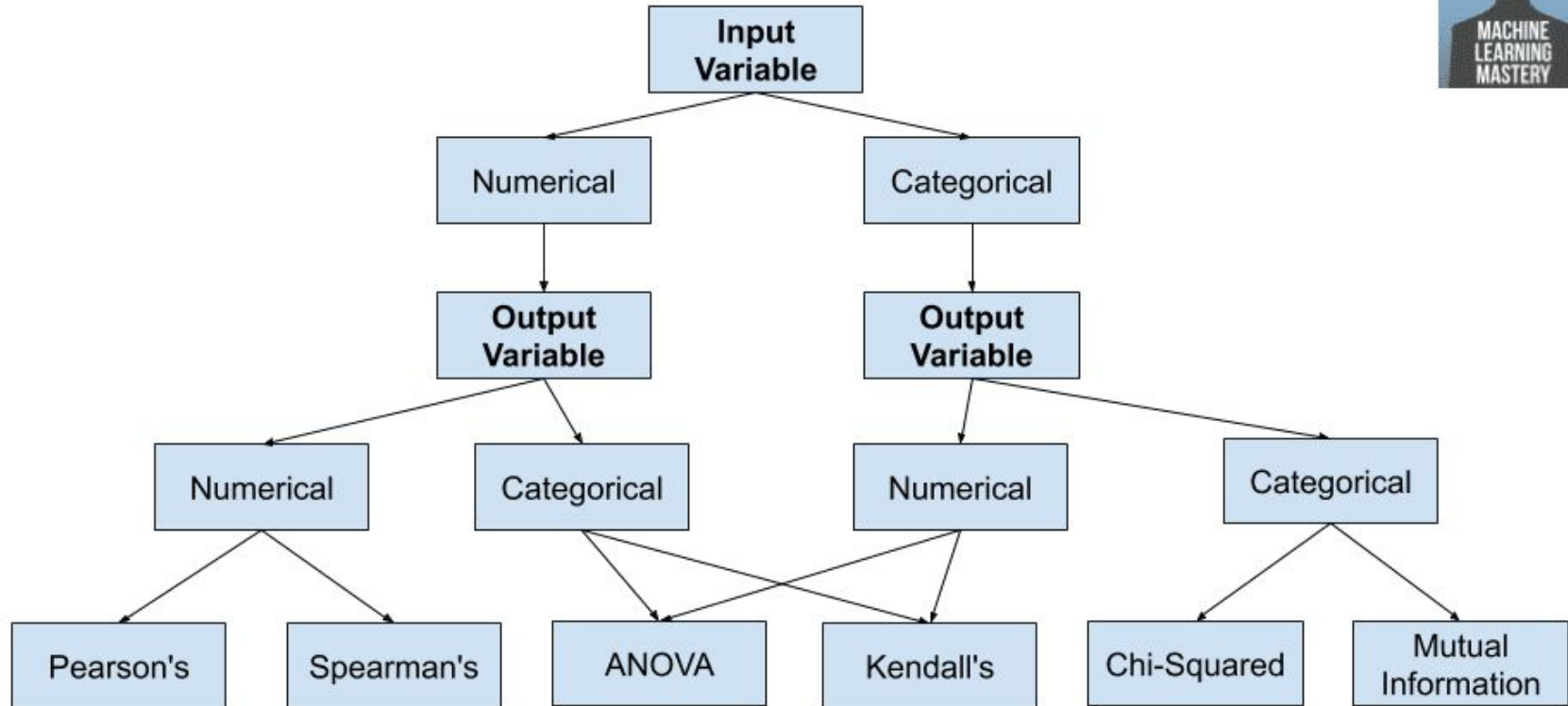
Overview of Feature Selection Techniques



Overview of Data Variable Types

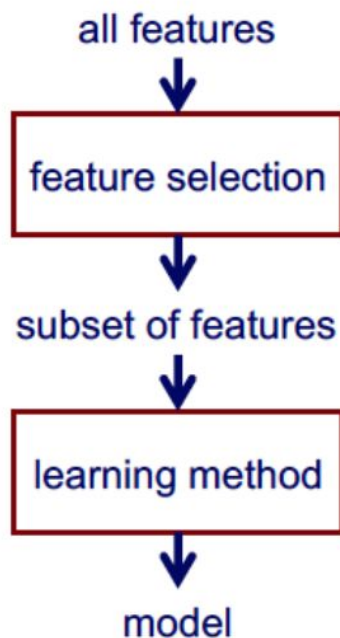


How to Choose a Feature Selection Method

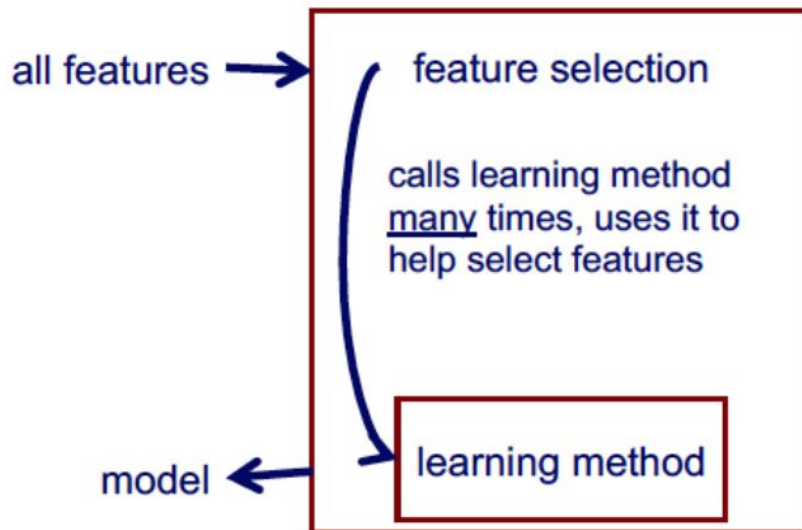


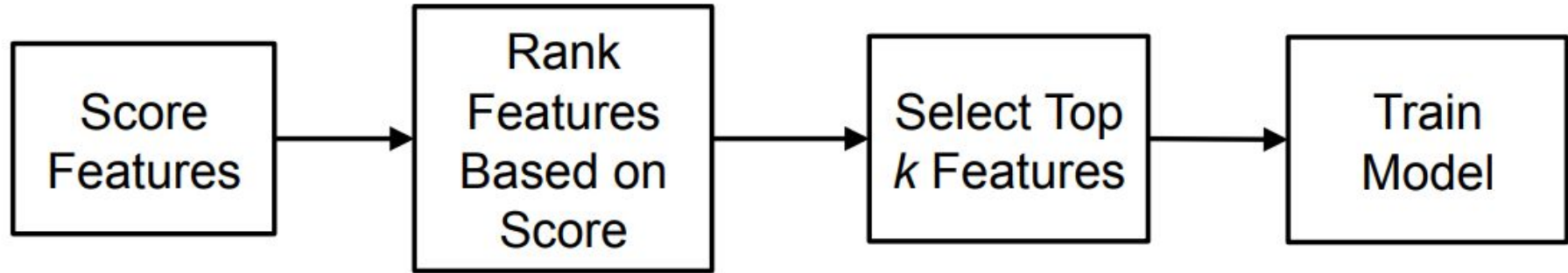
Feature seçimi ayrı görev olarak ele almak

filtering-based feature selection



wrapper-based feature selection





- Bu aşamada doğrulama yapılmadığından puanlar tahmin performansını temsil etmez.
- Validasyon ve test seti hesaplama için kullanılmaz.

- k değeri heuristic seçilir.
- Bir threshold belirlemek için pratik kurallar kullanılabilir. (örneğin, istatistiksel olarak anlamlı puanlara sahip özellikleri kullanın)
- Optimal bir k değeri seçmek için cross-validation kullanılabilir.

Removing features with low variance

```
>>> from sklearn.feature_selection import VarianceThreshold
>>> X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
>>> sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
>>> sel.fit_transform(X)
array([[0, 1],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 0],
       [1, 1]])
```

>>>

Univariate feature selection

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.feature_selection import SelectKBest
>>> from sklearn.feature_selection import chi2
>>> X, y = load_iris(return_X_y=True)
>>> X.shape
(150, 4)
>>> X_new = SelectKBest(chi2, k=2).fit_transform(X, y)
>>> X_new.shape
(150, 2)
```

>>>



Teşekkürler!