Scikit-learn Teorik Bilgi ve Kavramlar

1. Makine Öğrenmesi Temelleri

1.1 Makine Öğrenmesi Nedir?

Makine öğrenmesi, bilgisayarların verilerden öğrenerek belirli görevleri yerine getirmesini sağlayan yapay zeka alt dalıdır. Geleneksel programlamadan farklı olarak, makine öğrenmesi algoritmaları açık kurallar yerine veri örneklerinden öğrenir.

1.2 Öğrenme Türleri

Supervised Learning (Gözetimli Öğrenme)

- **Tanım**: Etiketli verilerle eğitim
- **Amaç**: Girdi-çıktı ilişkisini öğrenmek
- **Türler**:
  - **Classification**: Kategorik çıktı (örn: spam/not spam)
  - **Regression**: Sürekli çıktı (örn: ev fiyatı)

Unsupervised Learning (Gözetimsiz Öğrenme)

- **Tanım**: Etiketsiz verilerle eğitim
- **Amaç**: Veri yapısını keşfetmek
- **Türler**:
  - **Clustering**: Benzer veri noktalarını gruplamak
  - **Dimensionality Reduction**: Boyut azaltma
  - **Association**: İlişki kuralları bulma

Semi-supervised Learning (Yarı Gözetimli Öğrenme)

- **Tanım**: Hem etiketli hem etiketsiz verilerle eğitim
- **Amaç**: Az etiketli veriyle daha iyi performans

1.3 Overfitting ve Underfitting

Overfitting (Aşırı Öğrenme)

- **Tanım**: Model eğitim verisini ezberler, genelleme yapamaz
- **Belirtiler**: Eğitim hatası çok düşük, test hatası yüksek
- **Çözümler**:
  - Daha fazla veri toplama
  - Regularization (düzenlileştirme)
  - Cross-validation
  - Ensemble methods

Underfitting (Eksik Öğrenme)

- **Tanım**: Model veriyi yeterince öğrenemez
- **Belirtiler**: Hem eğitim hem test hatası yüksek
- **Çözümler**:
  - Model karmaşıklığını artırma
  - Daha fazla feature ekleme
  - Regularization azaltma

2. Linear Models Teorisi

2.1 Linear Regression

Matematiksel Formül

```
y = β₀ + β₁x₁ + β₂x₂ + ... + βₙxₙ + ε
```

Loss Function (MSE)

```
MSE = (1/n) * Σ(yᵢ - ŷᵢ)²
```

Normal Equation

```
β = (X^T X)^(-1) X^T y
```

2.2 Regularization

Ridge Regression (L2)

- **Formül**: `y = β₀ + β₁x₁ + ... + λΣβᵢ²`
- **Amaç**: Büyük katsayıları cezalandırır
- **Etki**: Smoothing, multicollinearity azaltma

Lasso Regression (L1)

- **Formül**: `y = β₀ + β₁x₁ + ... + λΣ|βᵢ|`
- **Amaç**: Sıfır katsayılar oluşturur
- **Etki**: Feature selection, sparse solutions

Elastic Net

- **Formül**: `y = β₀ + β₁x₁ + ... + λ₁Σ|βᵢ| + λ₂Σβᵢ²`
- **Amaç**: L1 ve L2'nin kombinasyonu

2.3 Logistic Regression

Sigmoid Function

```
σ(z) = 1 / (1 + e^(-z))
```

Log Loss

```
L = -Σ[yᵢ log(ŷᵢ) + (1-yᵢ) log(1-ŷᵢ)]
```

3. Support Vector Machines (SVM)

3.1 Linear SVM

Margin Concept

- **Margin**: En yakın veri noktaları arasındaki mesafe
- **Amaç**: Maksimum margin bulmak

Mathematical Formulation

```
min (1/2)||w||²
subject to: yᵢ(w^T xᵢ + b) ≥ 1
```

3.2 Kernel Trick

Linear Kernel

```
K(xᵢ, xⱼ) = xᵢ^T xⱼ
```

RBF Kernel

```
K(xᵢ, xⱼ) = exp(-γ||xᵢ - xⱼ||²)
```

Polynomial Kernel

```
K(xᵢ, xⱼ) = (γxᵢ^T xⱼ + r)^d
```

4. Decision Trees

4.1 Tree Structure

- **Root Node**: Başlangıç noktası
- **Internal Nodes**: Karar noktaları
- **Leaf Nodes**: Sonuç noktaları

4.2 Splitting Criteria

Gini Index

```
Gini = 1 - Σ(pᵢ²)
```

Entropy

```
Entropy = -Σ(pᵢ log₂(pᵢ))
```

Information Gain

```
IG = Entropy(parent) - Σ(|Sᵥ|/|S|) * Entropy(Sᵥ)
```

4.3 Pruning

- **Pre-pruning**: Ağaç büyürken durdurma
- **Post-pruning**: Ağaç tamamlandıktan sonra budama

5. Ensemble Methods

5.1 Bagging (Bootstrap Aggregating)

Bootstrap Sampling

- **Yöntem**: Random sampling with replacement
- **Amaç**: Çeşitlilik yaratmak

Aggregation

- **Classification**: Majority voting
- **Regression**: Averaging

5.2 Random Forest

Key Features

- **Bootstrap sampling**: Her ağaç farklı veri

- **Feature sampling**: Her split farklı feature subset
- **Ensemble**: Çok sayıda ağaç

Advantages

- Overfitting'e karşı dirençli
- Feature importance
- Out-of-bag estimation

5.3 Boosting

AdaBoost

- **Idea**: Zayıf öğrenicileri birleştir
- **Weighting**: Hatalı örneklerin ağırlığını artır

Gradient Boosting

- **Idea**: Residual'ları tahmin et
- **Loss function**: Gradient descent

6. Clustering Algorithms

6.1 K-Means

Algorithm

1. K centroid'i rastgele seç
2. Her noktayı en yakın centroid'e ata
3. Centroid'leri güncelle
4. 2-3'ü tekrarla

Objective Function

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

6.2 DBSCAN

Core Points

- **Definition**: MinPts komşusu olan noktalar
- **Border Points**: Core point'e komşu ama core olmayan
- **Noise Points**: Ne core ne border

Algorithm

1. Core point'leri bul

2. Core point'leri birleştir
3. Border point'leri ata

7. Dimensionality Reduction

7.1 Principal Component Analysis (PCA)

Mathematical Foundation

- **Covariance Matrix**: `C = (1/n) X^T X`
- **Eigenvectors**: Principal components
- **Eigenvalues**: Explained variance

Steps

1. Center the data
2. Compute covariance matrix
3. Find eigenvectors/eigenvalues
4. Project data

7.2 t-SNE

Objective

- **High-dimensional**: Similar points close
- **Low-dimensional**: Similar points close
- **Dissimilar points**: Far apart

Perplexity

- **Definition**: Effective number of neighbors
- **Typical values**: 5-50

8. Model Evaluation

8.1 Classification Metrics

Confusion Matrix

```
                Predicted
Actual    Positive  Negative
Positive    TP         FN
Negative    FP         TN
```

Metrics

- **Accuracy**: (TP + TN) / (TP + TN + FP + FN)
- **Precision**: TP / (TP + FP)

- **Recall**: TP / (TP + FN)
- **F1-Score**: 2 * (Precision * Recall) / (Precision + Recall)

ROC Curve

- **X-axis**: False Positive Rate
- **Y-axis**: True Positive Rate
- **AUC**: Area Under Curve

8.2 Regression Metrics

Mean Squared Error (MSE)

```
MSE = (1/n) Σ(yᵢ - ŷᵢ)²
```

Root Mean Squared Error (RMSE)

```
RMSE = √MSE
```

R^2 Score

```
R² = 1 - (SS_res / SS_tot)
```

8.3 Cross-Validation

K-Fold CV

- **Process**: Veriyi K parçaya böl
- **Training**: K-1 parça ile eğit
- **Validation**: 1 parça ile test
- **Repeat**: K kez

Stratified K-Fold

- **Purpose**: Class distribution koru
- **Use**: Classification problems

9. Feature Engineering

9.1 Feature Scaling

StandardScaler

```
z = (x - μ) / σ
```

MinMaxScaler

```
x_scaled = (x - x_min) / (x_max - x_min)
```

RobustScaler

```
x_scaled = (x - median) / IQR
```

9.2 Feature Selection

Filter Methods

- **Variance Threshold**: Düşük varyanslı features
- **Correlation**: Yüksek korelasyonlu features
- **Mutual Information**: Feature-target ilişkisi

Wrapper Methods

- **Forward Selection**: Sırayla feature ekle
- **Backward Elimination**: Sırayla feature çıkar
- **Recursive Feature Elimination**: Recursive elimination

Embedded Methods

- **Lasso**: L1 regularization
- **Ridge**: L2 regularization
- **Elastic Net**: L1 + L2

10. Hyperparameter Tuning

10.1 Grid Search

- **Method**: Tüm kombinasyonları dene
- **Pros**: Exhaustive search
- **Cons**: Computationally expensive

10.2 Random Search

- **Method**: Random combinations
- **Pros**: Faster, often better
- **Cons**: May miss optimal

10.3 Bayesian Optimization

- **Method**: Probabilistic model
- **Pros**: Efficient, adaptive
- **Cons**: Complex implementation

11. Bias-Variance Tradeoff

11.1 Bias

- **Definition**: Model assumptions vs reality
- **High Bias**: Underfitting
- **Low Bias**: Complex models

11.2 Variance

- **Definition**: Model sensitivity to data
- **High Variance**: Overfitting
- **Low Variance**: Simple models

11.3 Tradeoff

- **Simple Model**: High bias, low variance
- **Complex Model**: Low bias, high variance
- **Optimal**: Balance point

12. Practical Considerations

12.1 Data Quality

- **Missing Values**: Imputation strategies
- **Outliers**: Detection and handling
- **Noise**: Filtering techniques

12.2 Computational Efficiency

- **Algorithm Complexity**: Time and space
- **Scalability**: Large datasets
- **Parallelization**: Multi-core processing

12.3 Interpretability

- **Linear Models**: Coefficients
- **Trees**: Feature importance
- **Black Box**: SHAP, LIME

13. Advanced Topics

13.1 Transfer Learning

- **Concept**: Pre-trained models

- **Applications**: Computer vision, NLP
- **Benefits**: Less data, better performance

13.2 Active Learning

- **Concept**: Selective data labeling
- **Strategy**: Uncertainty sampling
- **Benefits**: Cost reduction

13.3 Multi-task Learning

- **Concept**: Multiple related tasks
- **Benefits**: Shared representations
- **Applications**: Recommendation systems

14. Best Practices

14.1 Data Preprocessing

1. Handle missing values
2. Remove outliers
3. Scale features
4. Encode categorical variables

14.2 Model Selection

1. Start with simple models
2. Use cross-validation
3. Compare multiple algorithms
4. Consider ensemble methods

14.3 Evaluation

1. Use appropriate metrics
2. Avoid data leakage
3. Test on unseen data
4. Monitor for drift

14.4 Deployment

1. Save preprocessing steps
2. Version control models
3. Monitor performance
4. Plan for updates