UNIVERSITY of WEST FLORIDA

Machine Learning

Module 4
Data Preprocessing

Brian Jalaian, Ph.D.
Associate Professor
Intelligent Systems & Robotics Department

Outline

UNIVERSITY of WEST FLORIDA

i. Introduction to Preprocessing
   A. Overview of preprocessing techniques in machine learning
   B. Importance of preprocessing for improving model performance

ii. II. Dealing with Missing Data
   A. Overview of missing data
   B. Different techniques for handling missing data
   C. Pros and cons of different techniques

iii. Handling Categorical Data
   A. Overview of categorical data
   B. Techniques for encoding categorical data
   C. Pros and cons of different encoding techniques

iv. Partitioning a Dataset into Training and Test Datasets
   A. Overview of train/test split
   B. How to split a dataset into training and test sets

v. Feature Scaling
   A. Overview of feature scaling
   B. Techniques for scaling features
   C. Pros and cons of different scaling techniques

vi. Selecting Meaningful Features
   A. Overview of feature selection
   B. Techniques for feature selection
   C. L1 and L2 regularization
   D. Geometric interpretation of L2 regularization
   E. Sparse solution with L1 regularization

vii. Assessing Feature Importance with Random Forests
   A. Overview of feature importance
   B. How to assess feature importance using random forests
   C. Interpretation of feature importance scores

Introduction to Preprocessing in Machine Learning

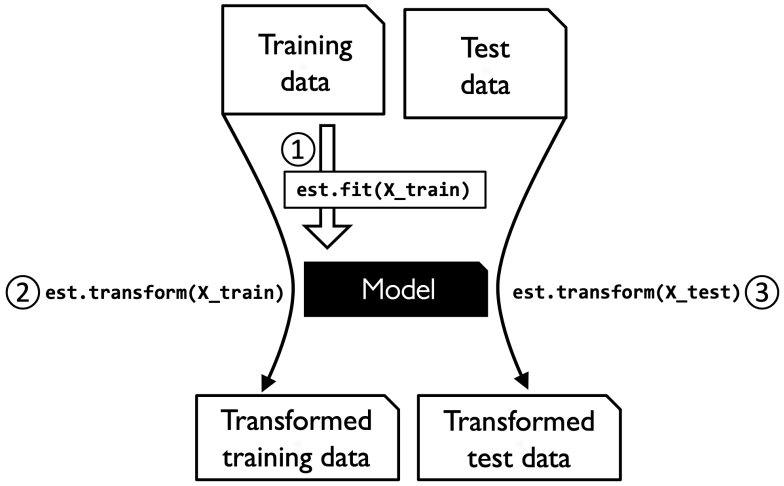
UNIVERSITY of WEST FLORIDA

- Introduction to preprocessing in machine learning
- Overview of preprocessing techniques
- Importance of preprocessing for improving model performance
- Common preprocessing techniques:
  - Dealing with missing data
  - Handling categorical data
  - Partitioning a dataset into separate training and test datasets
  - Bringing features on to the same scale
  - Selecting meaningful features
  - Assessing feature importance with random forests

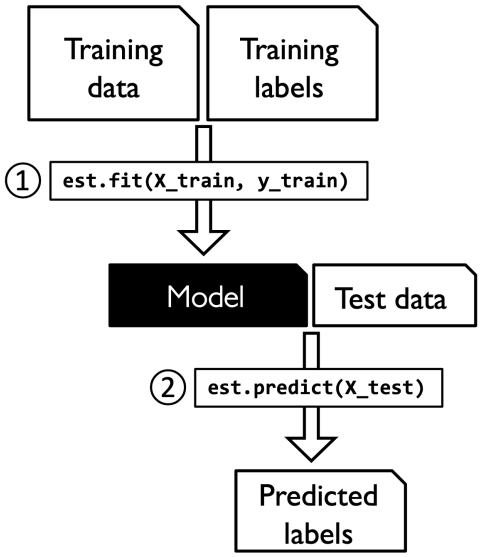
Understanding the scikit-learn Estimator API

UNIVERSITY of WEST FLORIDA

- Introduction to the scikit-learn Estimator API
- Overview of the Transformer API and its two essential methods: fit and transform
- Explanation of how the fit method is used to learn the parameters from the training data and the transform method uses those parameters to transform the data
- Requirement that the data array to be transformed must have the same number of features as the data array used to fit the model
- Overview of the Classifier API in scikit-learn
- Explanation of how the Classifier API belongs to the so-called Estimator in scikit-learn with an API conceptually similar to the Transformer API
- Discussion of how the fit method is used to learn the parameters of a model when training Classifiers for classification
- Explanation of how Classifiers in supervised classification tasks require class labels for fitting the model
- Overview of the predict method and its use in making predictions about new, unlabeled data examples



Using scikit-learn API for data transformation



Using scikit-learn API for predictive models such as classifiers

Dealing with missing data

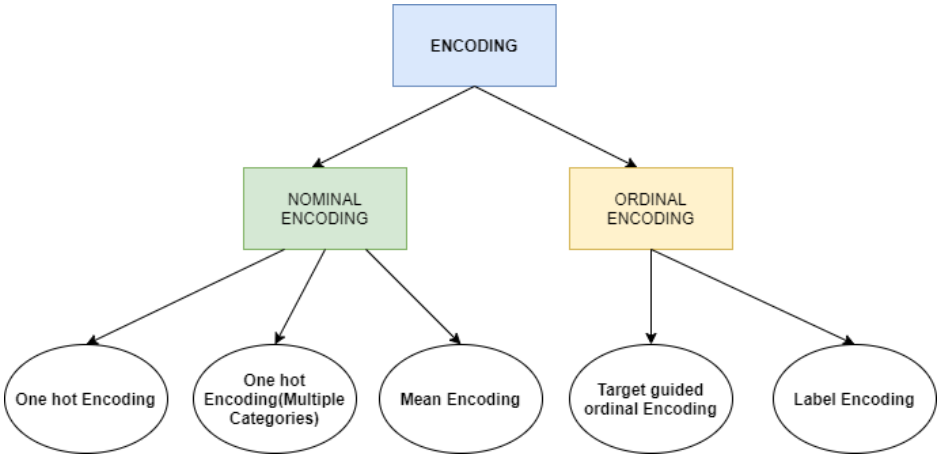
UNIVERSITY of WEST FLORIDA

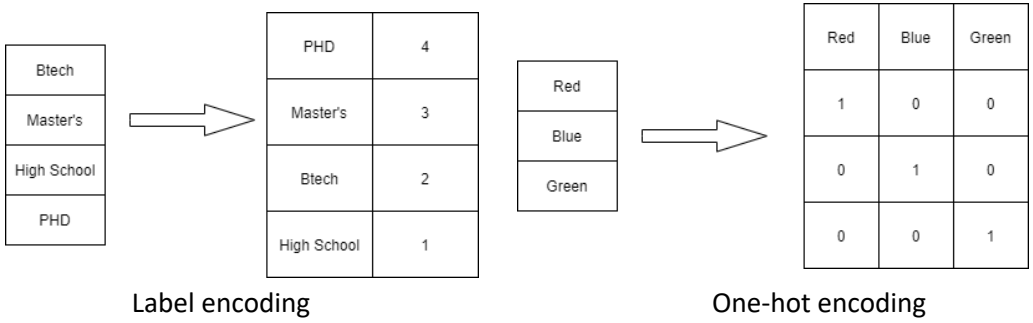
- Overview of the problem: Missing data can cause problems in machine learning as it can reduce sample size and result in bias.
- Different strategies for dealing with missing data:
  - Deletion: removing samples with missing data
  - Mean imputation: replacing missing values with the mean of the feature
  - Median imputation: replacing missing values with the median of the feature
  - etc.
- Pros and cons of each strategy
- Transitions to Jupyter code:
  - Identifying missing values in tabular data
  - Eliminating training examples or features with missing values
  - Imputing missing values

Handling Categorical Data

Transforming Non-Numerical Data for Machine Learning

- Explanation of Categorical Data
  - Variables with limited, fixed number of possible values
  - Represent categories or labels
- Importance of Handling Categorical Data
  - Most algorithms require numerical input data
  - Categorical data needs to be transformed
- Techniques for Handling Categorical Data
  - One-hot encoding
  - Label encoding
  - Ordinal encoding
- Encoding with Pandas
  - Mapping ordinal features
  - Encoding class labels
  - One-hot coding for nominal features
  - Encoding original features





Label encoding

One-hot encoding

Partitioning a Dataset into Separate Training and Test Datasets

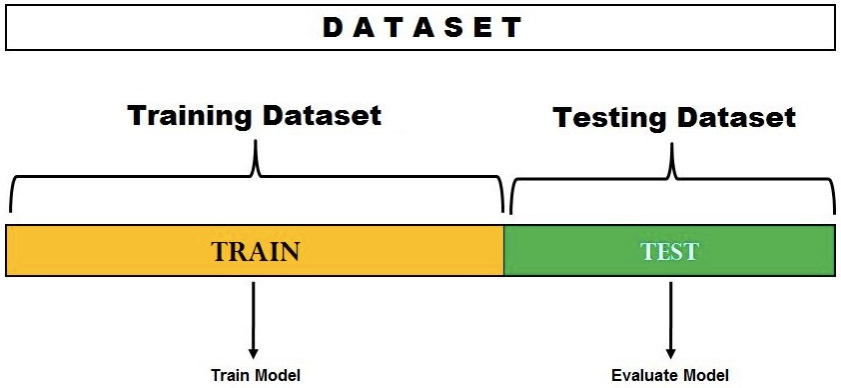
UNIVERSITY of WEST FLORIDA

I. Why Partition?
  - Evaluate model performance on unseen data
  - Prevent overfitting

II. Common Ratios of Division
  - 70% or 80% for training
  - 30% or 20% for testing

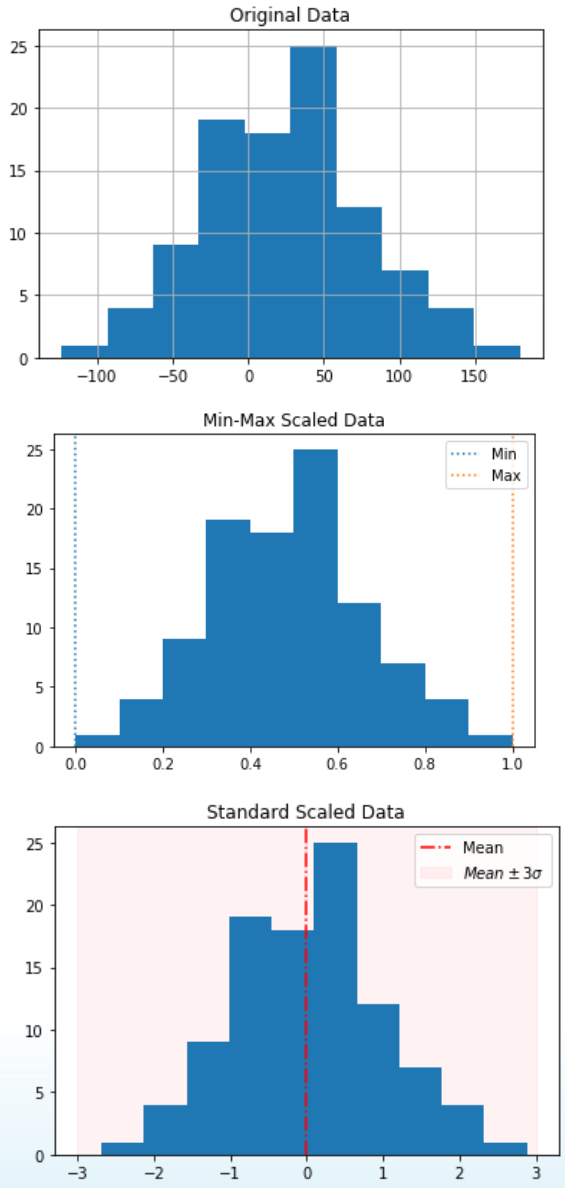
III. Partitioning with Popular Libraries
  - Scikit-learn's train_test_split function
  - Easy to partition with popular libraries

- IV. Demonstration of Code
  - Using train_test_split from scikit-learn



Feature Scaling

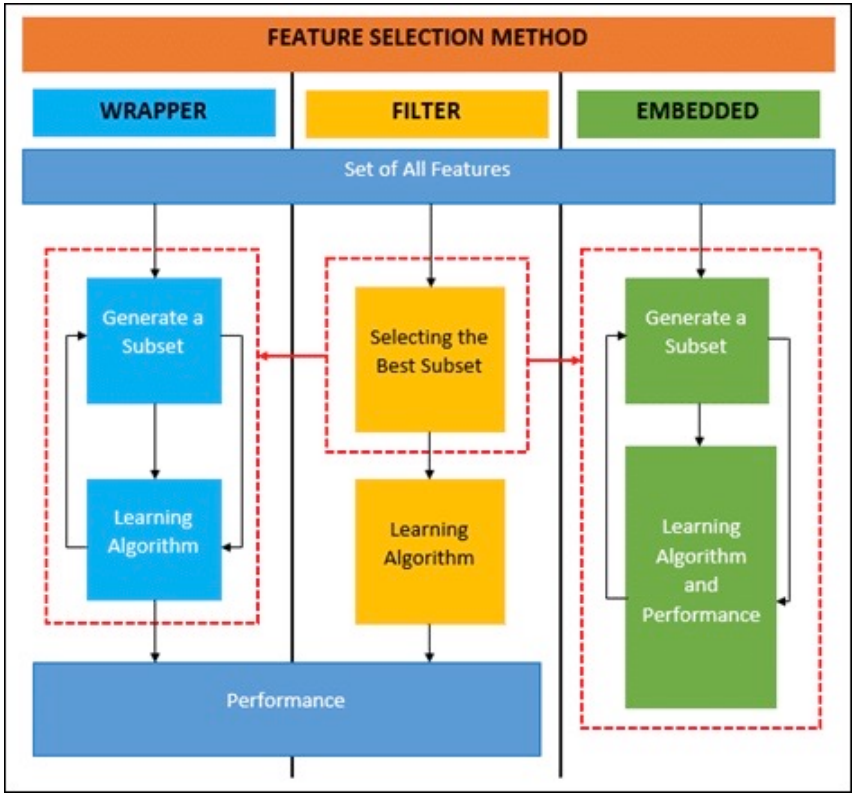
- Overview
  - Adjusting the scale of features to ensure similar ranges
  - Improves model performance and convergence speed
- Scaling Techniques
  - MinMaxScaler
  - StandardScaler
- Pros and Cons
  - MinMaxScaler: Rescales the data to a specific range, easy to interpret
  - StandardScaler: Centers the data around zero, robust to outliers



Selecting Meaningful Features

UNIVERSITY of WEST FLORIDA

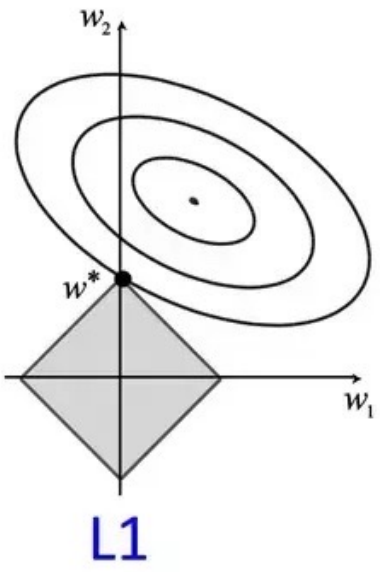
- Introduction
  - Reduce complexity and improve model performance
  - Overcome overfitting
  - Select the most informative features
- Methods
  - Filter methods: Statistical measures
  - Wrapper methods: Search algorithm
  - Embedded methods: Part of learning algorithm



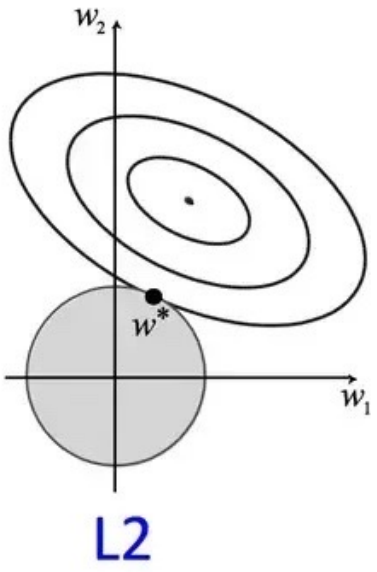
L1 and L2 Regularization as Penalties against Model Complexity

UNIVERSITY of WEST FLORIDA

- Introduction
  - What is Regularization
  - How Regularization helps reduce overfitting
  - L1 and L2 Regularization
- L1 Regularization
  - Definition L1: $||w||_1 = \sum_{j=1}^{m} |w_j|$
  - Penalty term
  - Resulting sparse solution
- L2 Regularization
  - Definition L2: $||w||_2^2 = \sum_{j=1}^{m} w_j^2$
  - Penalty term
  - Resulting dense solution



L1

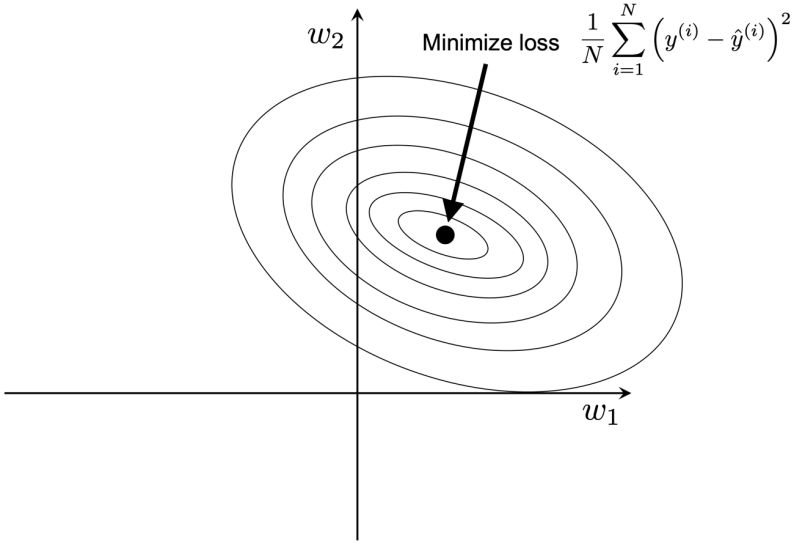


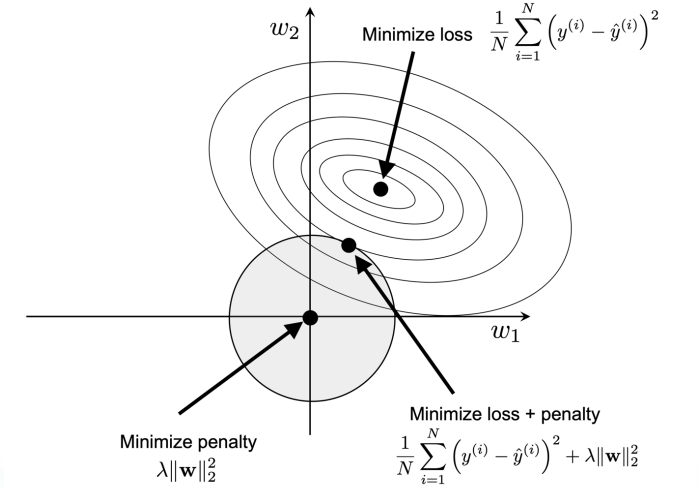
L2

Geometric Interpretation of L2 Regularization

UNIVERSITY of WEST FLORIDA

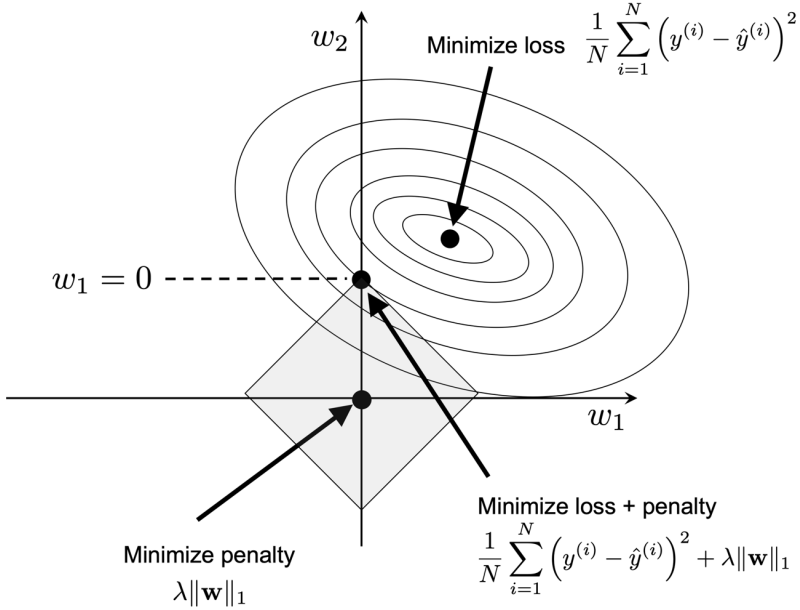
- Introduction
  - Explanation of L2 Regularization
  - Penalty against Model Complexity
- L2 Regularization as a Constraint
  - Explanation of L2 Regularization as a constraint on coefficient values
  - Visualization of the constraint as an elliptical region
  - Explanation of how the regularization term shrinks coefficients towards zero
- Effect on Model Complexity
  - Explanation of how L2 Regularization reduces overfitting
  - Comparison with other regularization techniques, such as L1 regularization and dropout





Sparse Solution with L1 Regularization

- Introduction
  - Explanation of L1 Regularization
  - Penalty against Model Complexity
- Sparse Solutions with L1 Regularization
  - Explanation of how L1 regularization can lead to sparse solutions, with many coefficients set to zero
- Comparison with Other Regularization Techniques
  - Comparison of L1 regularization with other regularization techniques, such as L2 regularization and dropout



Sequential Feature Selection

UNIVERSITY of WEST FLORIDA

- Introduction to Dimensionality Reduction
  - Explanation of reducing complexity and avoiding overfitting
- Types of Dimensionality Reduction
  - Feature Selection: selecting a subset of original features
  - Feature Extraction: deriving information from features to construct a new feature space
- Sequential Backward Selection Algorithm (SBS)
  - Explanation of SBS as a Feature Selection technique
  - Explanation of the SBS process, starting with all features and iteratively removing the feature with the lowest impact on the performance until a desired number of features is reached
- Pseudo Code for Sequential Backward Selection Algorithm

1. *Initialize the feature set with all features*
2. *Determine the performance metric to evaluate the feature set*
3. *While the desired number of features is not reached:*
   - *Evaluate the performance of each feature individually*
   - *Remove the feature with the lowest impact on performance*
4. *End while*

- Transition to Jupyter Code for Implementation

Popular Feature Selection Methods in scikit-learn

UNIVERSITY of WEST FLORIDA

- **Univariate Feature Selection:** Selects the best features based on univariate statistical tests.
- **Recursive Feature Elimination (RFE):** Recursively eliminates features and builds a model with the remaining features until the desired number of features is reached.
- **SelectFromModel:** Selects the best features based on the coefficients of a machine learning model.
- **SelectKBest:** Selects the best k features based on univariate statistical tests.
- **VarianceThreshold:** Selects features that have a minimum variance.

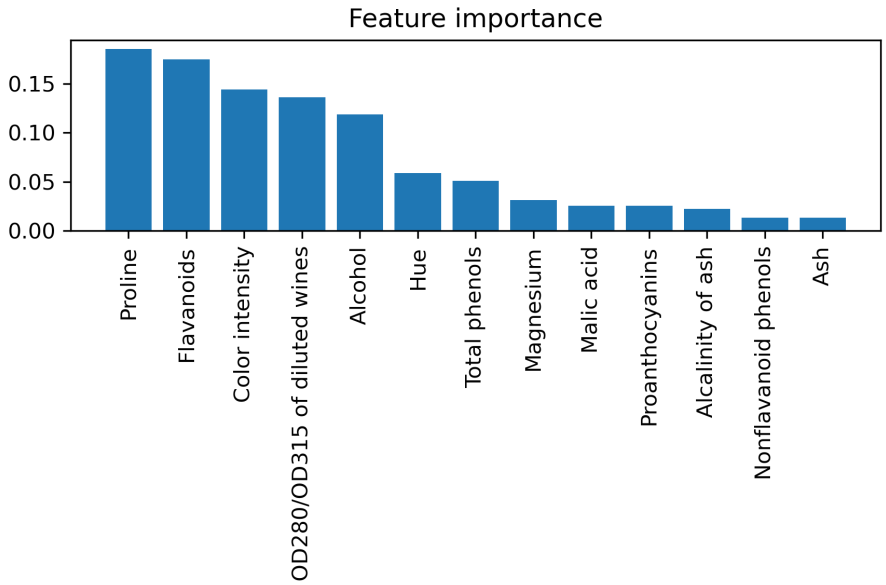
You can use the following scikit-learn functions for these techniques:

- Univariate Feature Selection: SelectKBest, SelectPercentile
- Recursive Feature Elimination: RFE
- SelectFromModel: SelectFromModel
- SelectKBest: SelectKBest
- VarianceThreshold: VarianceThreshold

Assessing Feature Importance with Random Forests

UNIVERSITY of WEST FLORIDA

- Introduction to Random Forest as an ensemble technique
- Explanation of measuring feature importance as the average impurity decrease computed from all decision trees in the forest
- Advantages of Random Forest in computing feature importance without making any assumptions about linear separability of the data
- Implementation of feature importance computation in scikit-learn through the feature_importances_ attribute after fitting the model
- Code demonstration of computing feature importance with Random Forest in Jupyter Notebook



Summary of Preprocessing Techniques in Machine Learning

UNIVERSITY of WEST FLORIDA

- Introduction to preprocessing techniques in machine learning and their importance for improving model performance
- Overview of techniques for dealing with missing data, such as mean imputation and median imputation, and their pros and cons
- Overview of techniques for encoding categorical data, such as one-hot encoding and label encoding, and their pros and cons
- Explanation of how to partition a dataset into training and test datasets using scikit-learn
- Overview of feature scaling techniques, such as MinMaxScaler and StandardScaler, and their pros and cons
- Overview of feature selection techniques, such as L1 and L2 regularization and sequential feature selection, and their impact on model complexity and overfitting
- Explanation of how to assess feature importance using random forests and interpretation of feature importance scores
- A summary of key takeaways from the module, including the importance of preprocessing techniques in machine learning and the various techniques for dealing with missing data, categorical data, scaling features, and selecting meaningful features.