Principal Component Analysis

By: **Aprameyan** and **Annamalai A**

Curse of Dimensionality

- Issues that arise when number of datapoints is small relative to the intrinsic dimension of the data.
- When dimensionality increases, the volume of the space increases so fast that the data becomes sparse.
- To obtain a good result, the amount of data increases exponentially.
- As number of features increases, data required to train models accurately grows rapidly.

Principal Component Analysis

Introduction

- Dimensionality Reduction Technique
- Applications:
    - Exploratory Data Analysis
    - Visualization
    - Data Preprocessing
- Dimensionality reduction:
    - Process of transforming data from a high-dimensional space to a low-dimensional space
    - Low-dimensional representation. Retains some meaningful properties of the original data, close to its intrinsic dimension
- Principal Component Analysis:
    - Technique for dimensionality reduction
    - Identifies a set of orthogonal axes, called principal components

    - Captures the maximum variance in the data

Principal components:

- In a collection of points in a real coordinate space
    - Sequence of p unit vectors
    - i^{th} vector is the direction of a line that best fits the data while being perpendicular to the vectors 0 to $i-1$
- Best fitting line is defined as the line that minimizes the "Average Squared Perpendicular Distance" from the points to the line

Recursive Feature Elimination (RFE)

Recursive feature elimination is a method used to eliminate features that don't matter to the model or don't contribute to the improvement of the model.

RFE selector is first trained on all features so that it knows about each feature in the data and the importance of it which is obtained by a specific attribute, such as weight, then it prunes the least important features.

This process is done recursively until the desired number of features remains, and all of these features contribute greatly to the improvement of the model.

Recursive Feature Elimination Cross Validation (RFECV)

What is RFECV?

RFECV stands for **Recursive Feature Elimination with Cross-Validation**.

It is a feature selection method used to automatically find the *best subset of features* for a machine learning model.

It removes weak features step-by-step and evaluates model performance after each removal using cross-validation.

Why do we use RFECV?

- To reduce overfitting by removing irrelevant features

- To improve model accuracy
- To reduce training time
- To automatically choose the optimal number of features
- To avoid manually guessing which features to drop

How does RFECV work?

RFECV performs the following steps:

1. Train a model on all features
2. Rank the features based on importance
3. Remove the weakest feature
4. Re-train the model
5. Evaluate using cross-validation
6. Repeat this process until the best performing feature set is found

This creates multiple "feature subsets" and picks the one with highest CV score.

How RFECV chooses the best features

RFECV uses **cross-validation score** (default = model accuracy for classifiers) to evaluate how good each subset is.

The subset with the **maximum CV score** is chosen as the final optimal feature set.

Algorithms that can be used with RFECV

You can use:

- Logistic Regression
- Linear Regression
- Decision Tree
- Random Forest
- SVM
- Any model that provides a feature importance or coefficie

Output Interpretation

- `n_features_`: number of selected best features
- `support_`: True/False list indicating which features are selected
- `ranking_`: 1 = best feature, higher numbers = weaker features

Advantages of RFECV

- Automatic feature selection
- Helps avoid overfitting
- Improves model generalization
- Works with any model that has coefficients or feature importances
- Uses cross-validation to ensure stability

Disadvantages of RFECV

- Can be slow for large datasets
- Computationally expensive for high-dimensional data
- Performance depends heavily on the chosen model
- May remove features that are only useful in combination

When should you use RFECV?

Use RFECV when:

- You want the model to tell you the best features
- You want cross-validation-based feature selection
- You are unsure which features matter
- You want to reduce dimensionality without PCA

Don’t use RFECV when:

- Dataset is extremely large
- Model training is slow
- You have thousands of features (use other methods like SelectKBest instead)

RFE vs RFECV

Feature Selection Method	RFE	RFECV
Full Form	Recursive Feature Elimination	Recursive Feature Elimination with Cross-Validation
Main Goal	Select features by recursively removing the weakest ones	Automatically select the best number of features using cross-validation
How it Works	Removes least important features step-by-step based on estimator coefficients/feature importance	Performs RFE, then evaluates each subset using CV, and finally selects best-performing subset
Needs Cross-Validation?	No	Yes
Automatic Selection of Optimal Number of Features?	No, you must manually specify `n_features_to_select`	Yes, RFECV finds the optimal number automatically
Speed	Faster	Slower
Computation Cost	Low to Medium	High
Ideal For	Smaller datasets where you know roughly how many features you need	Datasets where you want model-driven automatic feature selection
Risk	Might select too many or too few features	More reliable because it uses CV to choose best feature subset
Output	Feature ranking + support mask	Feature ranking + support mask + number of optimal features
sklearn Class	`sklearn.feature_selection.RFE`	`sklearn.feature_selection.RFECV`