Feature Selection

Feature Selection

Goal: Remove irrelevant or redundant features to make models simpler, faster, and more accurate.

1. Filter Methods (Based on Statistics)

- **Chi-Square**
- **Variance Threshold**
- **Correlation**

Chi-Square Test

Purpose: To test the dependence between two categorical features.

- Example: “Passed Exam?” vs. “Studied?” — Are they related?
- If not related, one of the features might be dropped.

Steps:

1. Create a contingency table.
2. Apply the Chi-Square formula or use a library function.
3. Check the **p-value**:
   - If $p < 0.05$, features are related.
   - If $p \geq 0.05$, they may be considered independent.

Python Example:

```
from sklearn.feature_selection import SelectKBest, chi2

# X: features (categorical, encoded as integers), y: target
selector = SelectKBest(score_func=chi2, k=2)
X_new = selector.fit_transform(X, y)
```

Variance Threshold

Purpose: Identify and remove features with very low variance.

- If a feature has almost the same value for all samples, it carries little information.

Steps:

1. Compute variance of each feature.
2. Drop features below a threshold.

Python Example:

```
from sklearn.feature_selection import VarianceThreshold

# Threshold of 0 means remove features with the same value in all
    samples
selector = VarianceThreshold(threshold=0.01)
X_new = selector.fit_transform(X)
```

Correlation

Purpose: Find highly correlated features that may be redundant.
Steps:

1. Compute the Pearson correlation coefficient (r) between each feature pair.
2. If $|r|$ is close to 1, consider dropping one of the features.

Python Example:

```
import pandas as pd
import numpy as np

corr_matrix = df.corr().abs()

# Select upper triangle of correlation matrix
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).
    astype(bool))

# Find features with correlation > 0.9
to_drop = [column for column in upper.columns if any(upper[column]
    > 0.9)]

df_reduced = df.drop(columns=to_drop)
```