Feature Selection and Importance

Why Feature Selection Matters

Interpretability + Efficiency

Selection Methods

Filter
Correlation-based

Wrapper
RFE (Recursive Feature Elimination)

Embedded
Lasso regularization

Balance: Model Simplicity ⚖ Predictive Power

Coefficient Magnitude
Indicates feature importance

Standardization
Standardize features before comparing coefficients

Beware
Multicollinearity affects importance interpretation

Permutation Importance
Alternative robust method

Example: Applying Feature Selection Methods

Filter Method

Step 1: Calculate Correlation

Compute correlation between each feature and target

Wrapper Method (RFE)

Step 1: Train Full Model

Fit model with all features

Embedded Method (Lasso)

Step 1: Set Regularization

Choose penalty parameter λ

```
corr_matrix = X.corrwith(y)
```

Step 2: Set Threshold

Select features with |correlation| > 0.5

```
threshold = 0.5
```

Step 3: Select Features

Keep features above threshold

```
selected = corr_matrix[abs(corr_matrix) > 0.5]
```

✓ Result

Fast, independent of model
Good for initial screening

```
model.fit(X_all, y)
```

Step 2: Rank Features

Identify least important feature

```
importance = model.coef_
```

Step 3: Remove & Repeat

Eliminate feature, retrain model

```
X_new = X.drop(worst_feature)
model.fit(X_new, y)
```

✓ Result

Model-specific selection
More accurate but slower

```
model = Lasso(alpha=0.1)
```

Step 2: Train Model

L1 penalty shrinks coefficients

```
model.fit(X, y)
```

Step 3: Extract Features

Select features with non-zero coefficients

```
selected = X.columns[model.coef_ != 0]
```

✓ Result

Automatic selection
Built into training process