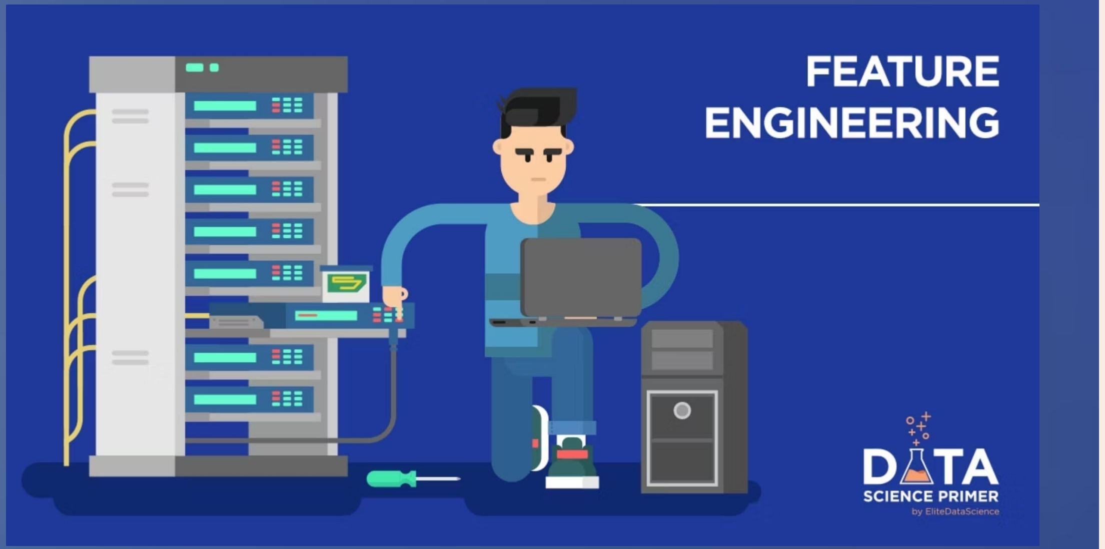


Feature Selection and Engineering in Machine Learning

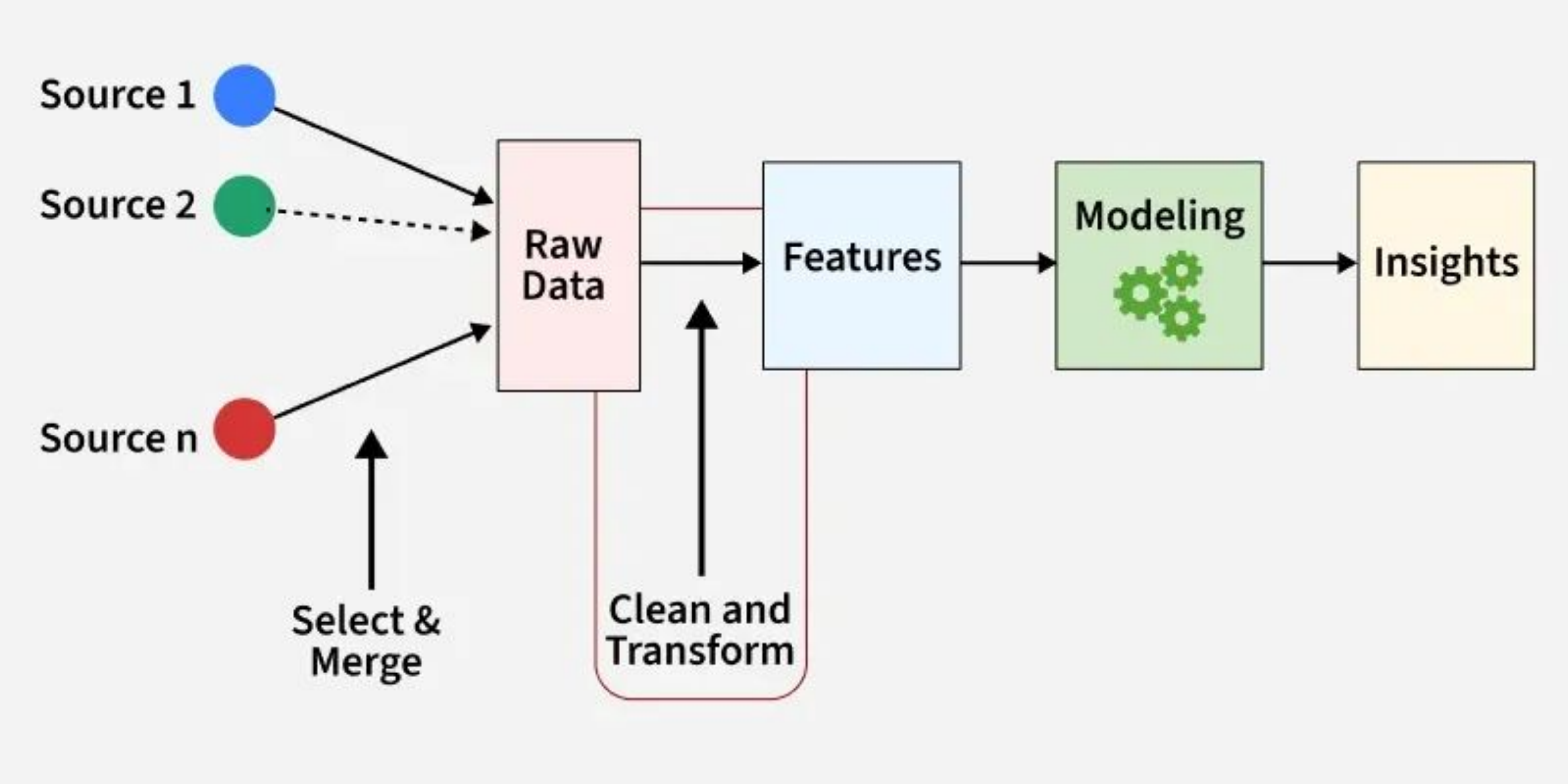
What is feature engineering?

The process of identifying and selecting the most important features (variables) from a dataset that contribute to the predictive power of a machine learning model.

1 **Feature engineering is essential for machine learning, transforming raw data into usable features.**

2 **The quality of features directly affects the accuracy of predictions made by algorithms.**

3 **The first step is feature creation, identifying and generating new feature from existing data.**


Source 1
Source 2
Source n
Select & Merge
Raw Data
Clean and Transform
Features
Modeling
Insights

1. **Feature Creation:** Feature creation involves generating new features from domain knowledge or by observing patterns in the data.
    a. **Domain-specific:** Created based on industry knowledge like business rules.
    b. **Data-driven:** Derived by recognizing patterns in data.
    c. **Synthetic:** Formed by combining existing features.

2. **Feature Transformation: Transformation adjusts features to improve model learning:**
    a. **Normalization & Scaling:** Adjust the range of features for consistency.
    b. **Encoding:** Converts categorical data to numerical form i.e one-hot encoding.
    c. **Mathematical transformations:** Like logarithmic transformations for skewed data.

3. **Feature Extraction:** Extracting meaningful features can reduce dimensionality and improve model accuracy:
    a. **Dimensionality reduction:** Techniques like PCA reduce features while preserving important information.
    b. **Aggregation & Combination:** Summing or averaging features to simplify the model.

4. **Feature Selection: Feature selection involves choosing a subset of relevant features to use:**
    a. **Filter methods:** Based on statistical measures like correlation.
    b. **Wrapper methods:** Select based on model performance.
    c. **Embedded methods:** Feature selection integrated within model training.

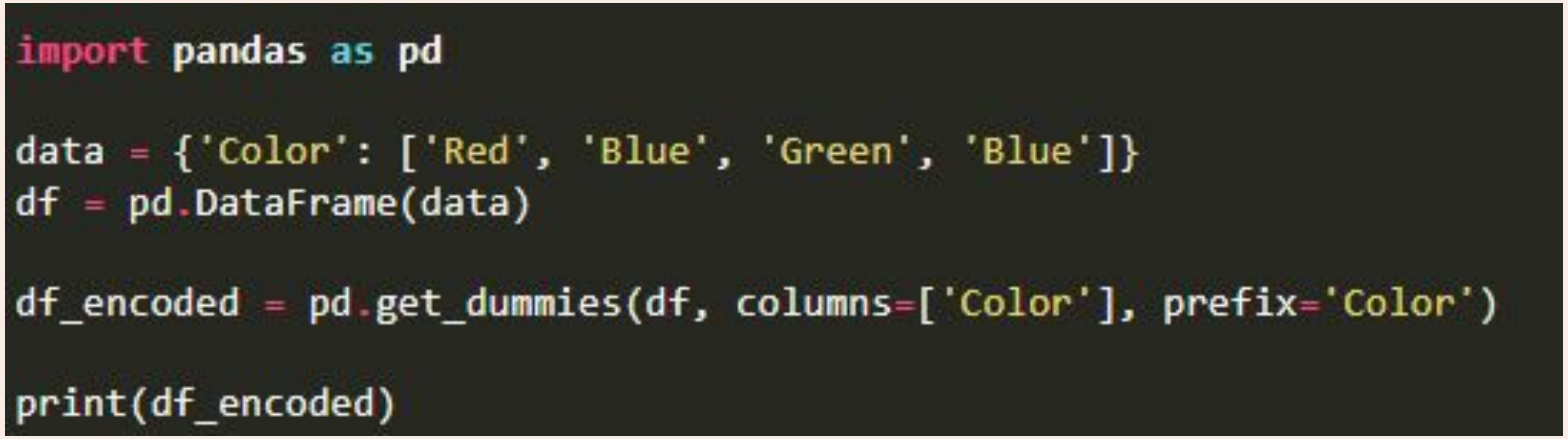
5. **Feature Scaling: Scaling ensures that all features contribute equally to the model:**
    a. **Min-Max scaling:** Rescales values to a fixed range like 0 to 1.
    b. **Standard scaling:** Normalizes to have a mean of 0 and variance of 1.

Steps in feature engineering:

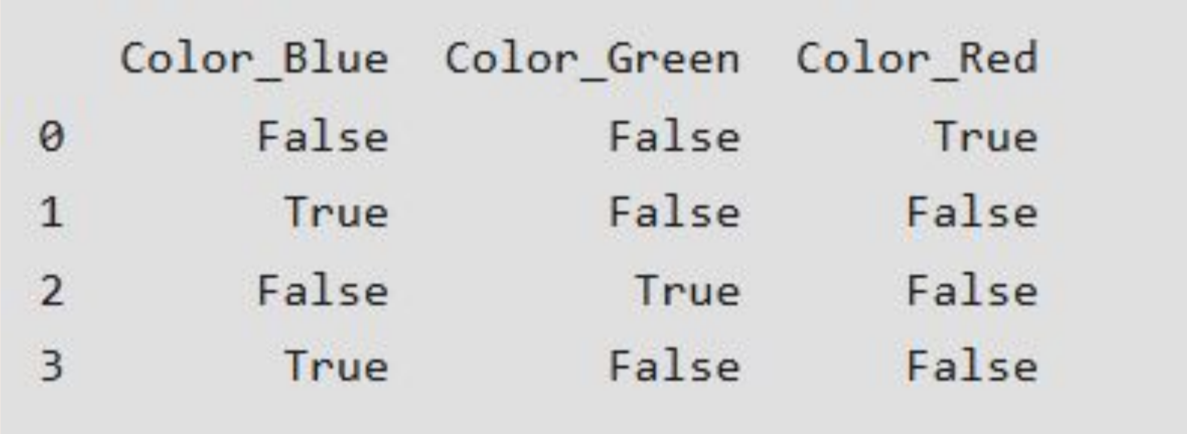
1. Data Cleaning: Identify and correct errors or inconsistencies in the dataset to ensure data quality and reliability.
2. Data Transformation: Transform raw data into a format suitable for modeling including scaling, normalization and encoding.
3. Feature Extraction: Create new features by combining or deriving information from existing ones to provide more meaningful input to the model.
4. Feature Selection: Choose the most relevant features for the model using techniques like correlation analysis, mutual information and stepwise regression.
5. Feature Iteration: Continuously refine features based on model performance by adding, removing or modifying features for improvement.

One Hot encoding:

One Hot Encoding is a method for converting categorical variables into a binary format. It creates new columns for each category where 1 means the category is present and 0 means it is not. The primary purpose of One Hot Encoding is to ensure that categorical data can be effectively used in machine learning models.



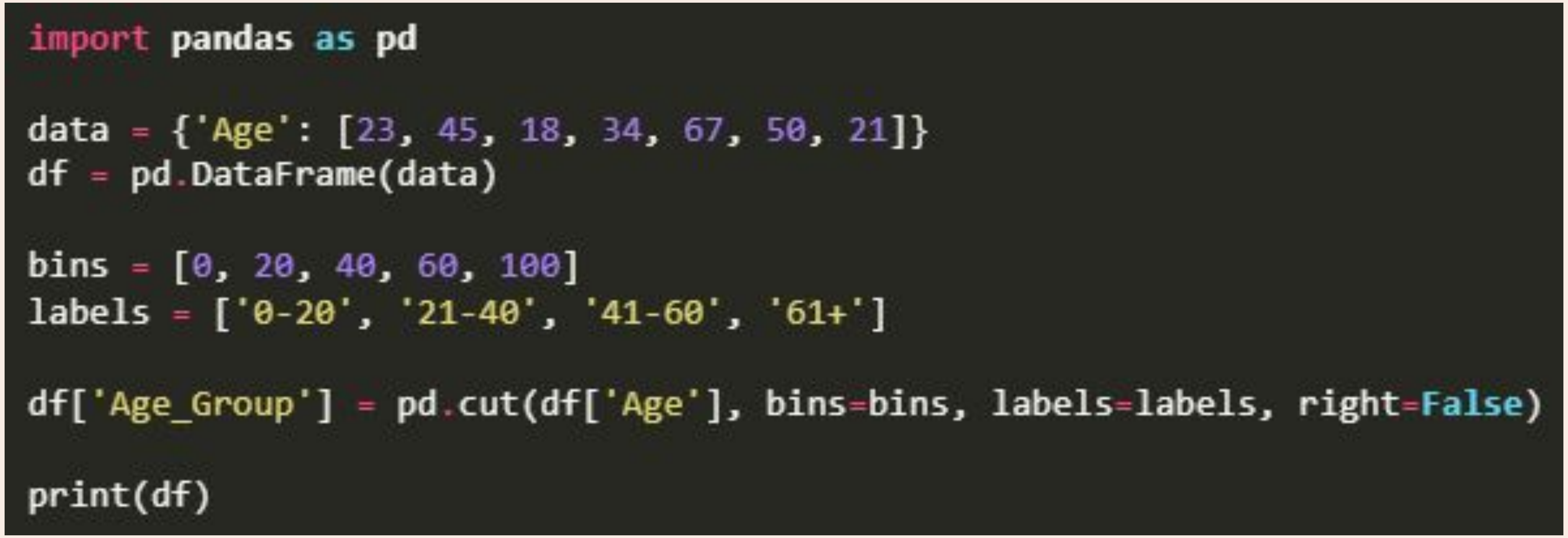
```
import pandas as pd

data = {'Color': ['Red', 'Blue', 'Green', 'Blue']}
df = pd.DataFrame(data)

df_encoded = pd.get_dummies(df, columns=['Color'], prefix='Color')

print(df_encoded)
```



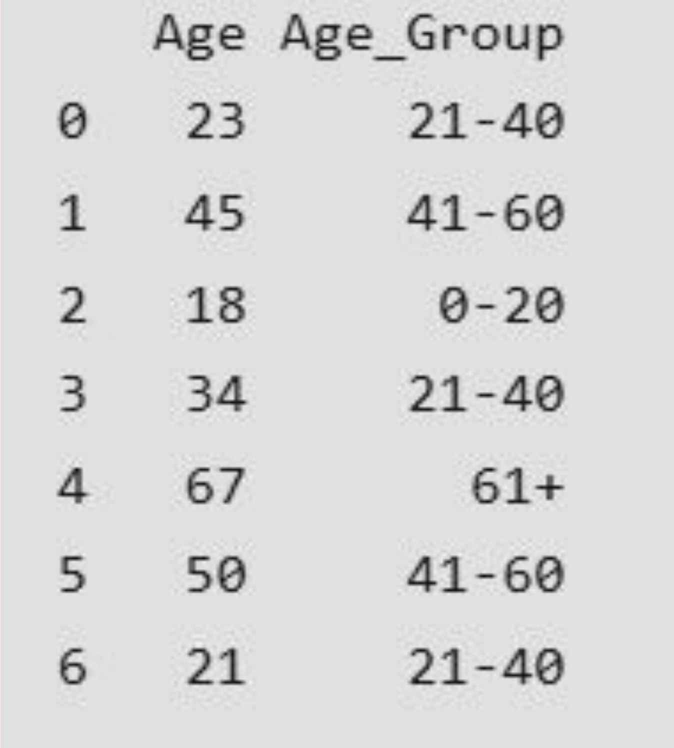
```
   Color_Blue  Color_Green  Color_Red
0       False        False       True
1        True        False      False
2       False         True      False
3        True        False      False
```

Binning:

Data binning or bucketing is a data preprocessing method used to minimize the effects of small observation errors. The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin. This has a smoothing effect on the input data and may also reduce the chances of overfitting in the case of small datasets



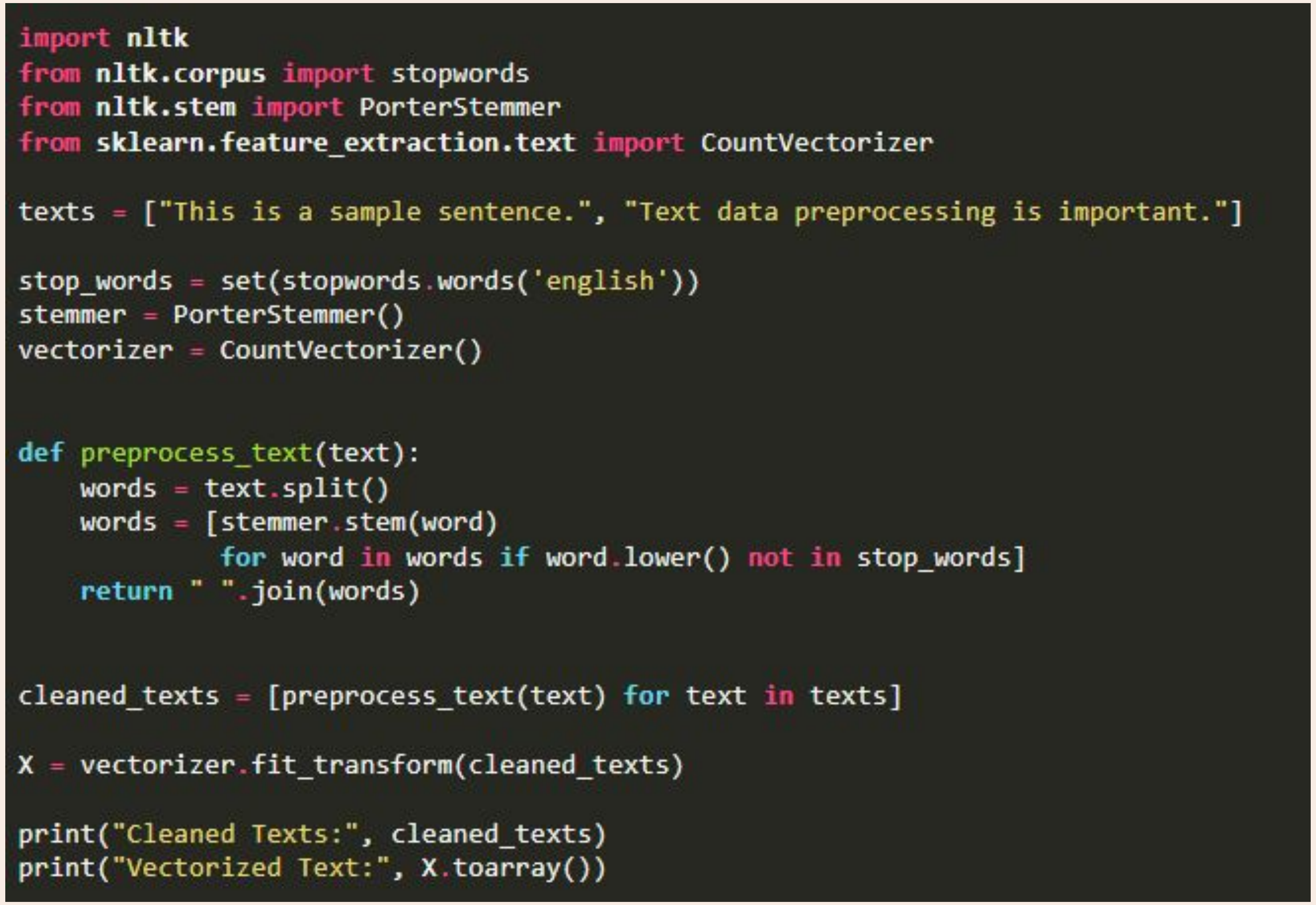
```python
import pandas as pd

data = {'Age': [23, 45, 18, 34, 67, 50, 21]}
df = pd.DataFrame(data)

bins = [0, 20, 40, 60, 100]
labels = ['0-20', '21-40', '41-60', '61+']

df['Age_Group'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

print(df)
```



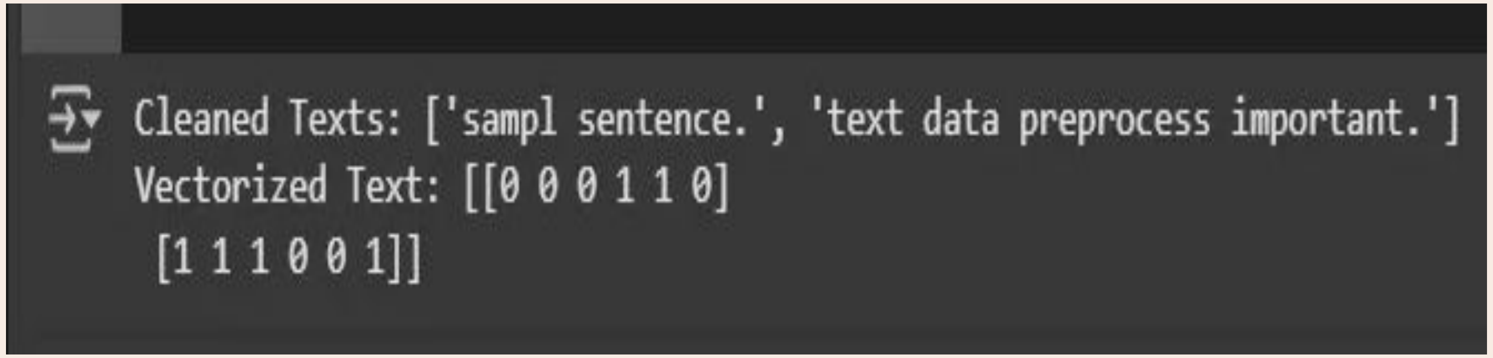
```
   Age Age_Group
0   23     21-40
1   45     41-60
2   18      0-20
3   34     21-40
4   67       61+
5   50     41-60
6   21     21-40
```

Text Preprocessing:

Natural Language Processing (NLP) has advanced significantly and now plays an important role in multiple real-world applications like chatbots, search engines and sentiment analysis. An early step in any NLP workflow is text preprocessing, which prepares raw textual data for further analysis and modeling.



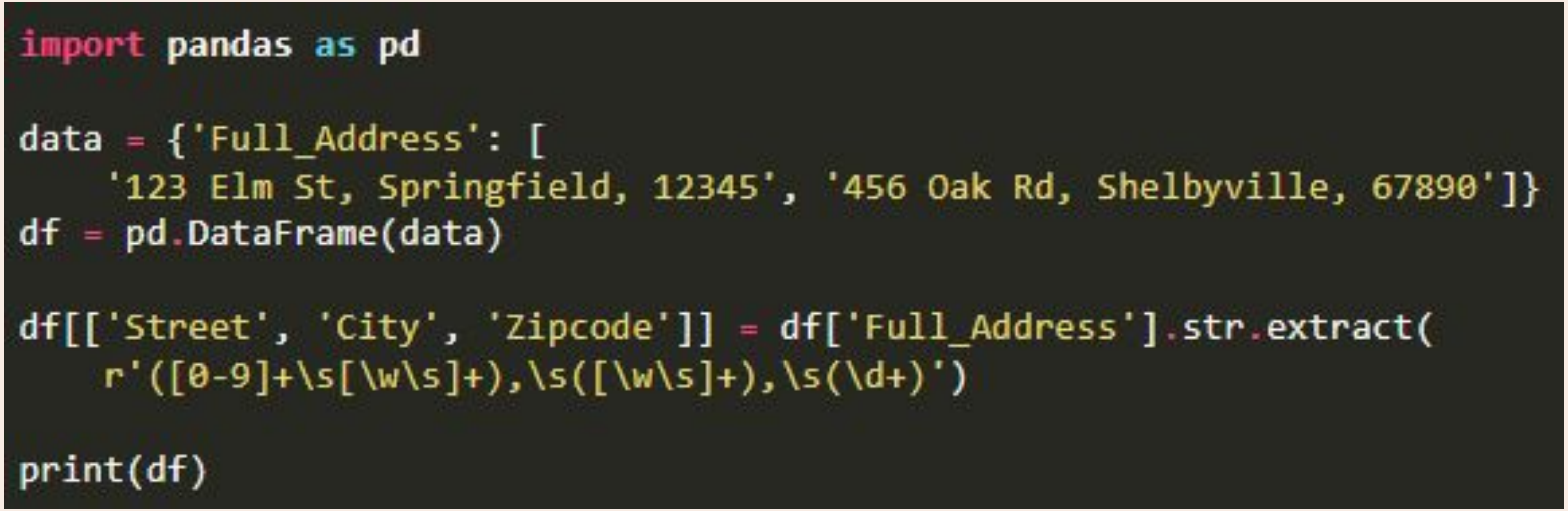
```python
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

texts = ["This is a sample sentence.", "Text data preprocessing is important."]

stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()
vectorizer = CountVectorizer()


def preprocess_text(text):
    words = text.split()
    words = [stemmer.stem(word)
             for word in words if word.lower() not in stop_words]
    return " ".join(words)


cleaned_texts = [preprocess_text(text) for text in texts]

X = vectorizer.fit_transform(cleaned_texts)

print("Cleaned Texts:", cleaned_texts)
print("Vectorized Text:", X.toarray())
```



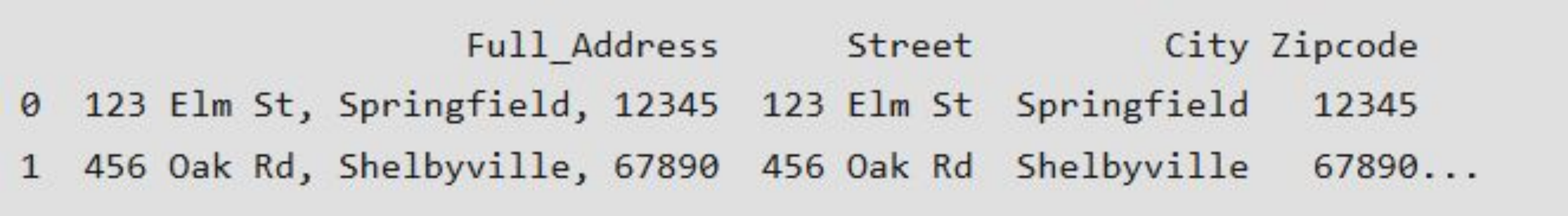
```
Cleaned Texts: ['sampl sentence.', 'text data preprocess important.']
Vectorized Text: [[0 0 0 1 1 0]
 [1 1 1 0 0 1]]
```

Feature Splitting:

Feature splitting is a concept used in data preprocessing and feature engineering, where a single feature (column) is divided into multiple sub-features to make the data more useful for machine learning models.



```
import pandas as pd

data = {'Full_Address': [
    '123 Elm St, Springfield, 12345', '456 Oak Rd, Shelbyville, 67890']}
df = pd.DataFrame(data)

df[['Street', 'City', 'Zipcode']] = df['Full_Address'].str.extract(
    r'([0-9]+\s[\w\s]+),\s([\w\s]+),\s(\d+)')

print(df)
```



```
                     Full_Address      Street         City Zipcode
0  123 Elm St, Springfield, 12345  123 Elm St  Springfield   12345
1  456 Oak Rd, Shelbyville, 67890  456 Oak Rd  Shelbyville   67890...
```

Tools in feature engineering:

1. **Featuretools:** Automates feature engineering by extracting and transforming features from structured data. It integrates well with libraries like pandas and scikit-learn making it easy to create complex features without extensive coding.
2. **TPOT:** Uses genetic algorithms to optimize machine learning pipelines, automating feature selection and model optimization. It visualizes the entire process, helping you identify the best combination of features and algorithms.
3. **DataRobot:** Automates machine learning workflows including feature engineering, model selection and optimization. It supports time-dependent and text data and offers collaborative tools for teams to efficiently work on projects.
4. **Alteryx:** Offers a visual interface for building data workflows, simplifying feature extraction, transformation and cleaning. It integrates with popular data sources and its drag-and-drop interface makes it accessible for non-programmers.
5. **H2O.ai:** Provides both automated and manual feature engineering tools for a variety of data types. It includes features for scaling, imputation and encoding and offers interactive visualizations to better understand model results.

What is feature selection?

The process of transforming raw data into meaningful features that better represent the underlying structure of the data, thus improving model performance.

1 **Improve model performance: Reduces overfitting and increases accuracy by eliminating unnecessary data.**

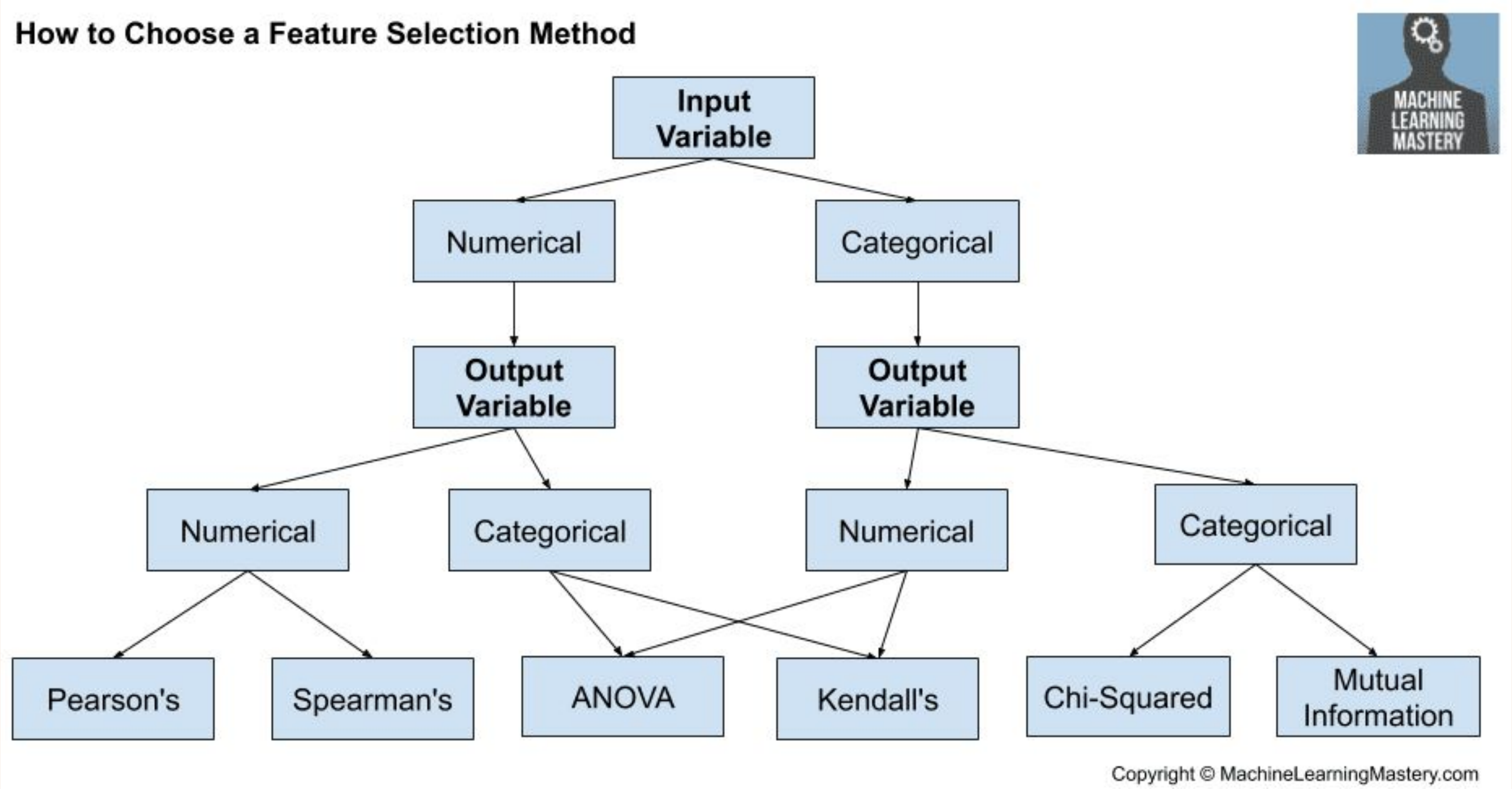
2 **Save resources: Lowers computational cost and memory usage.**

3 **Reduce complexity: Makes models simpler and faster to train and interpret.**

4 **Enhance generalization: Helps the model perform better on unseen data.**

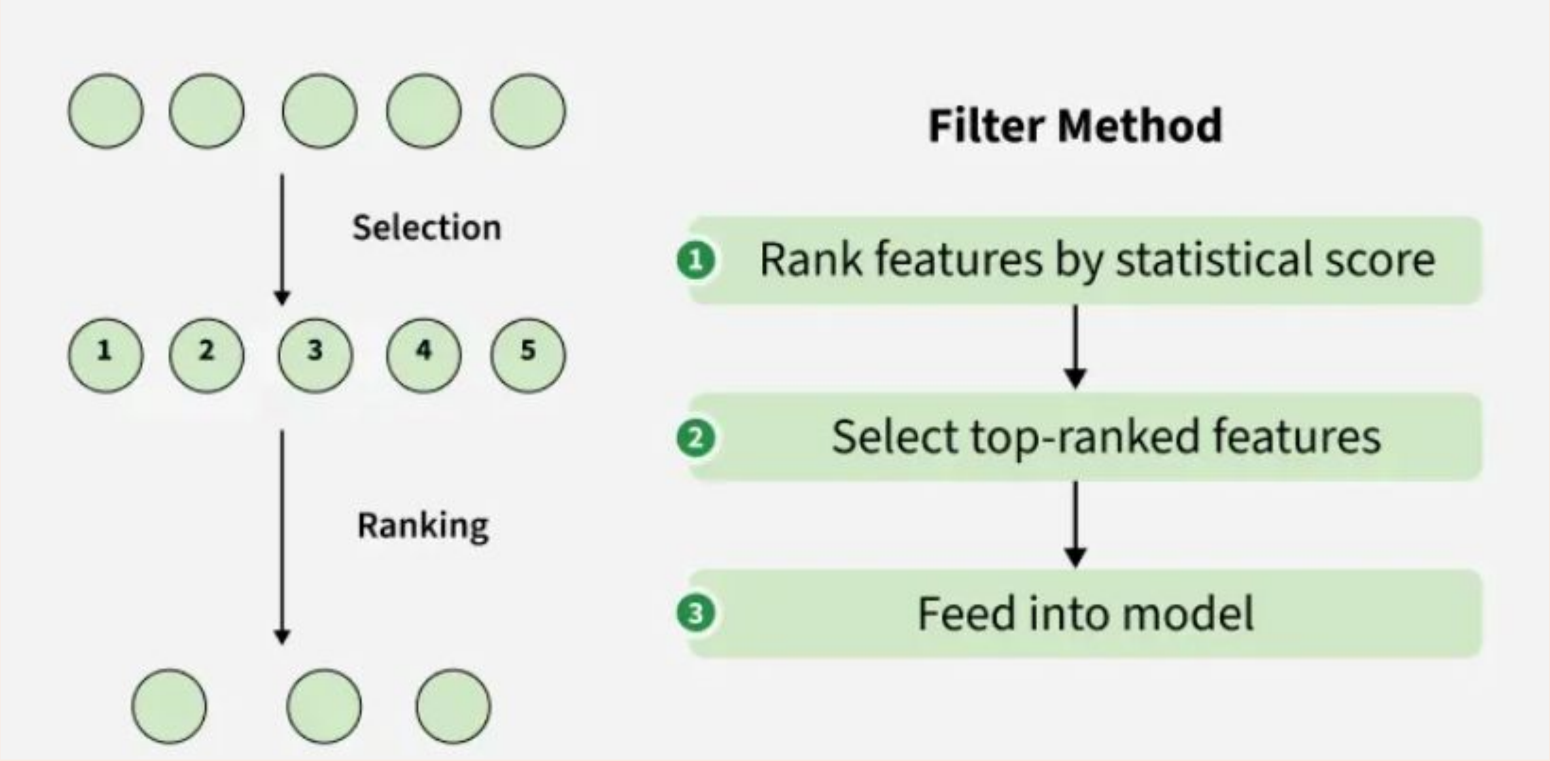
Feature selection:

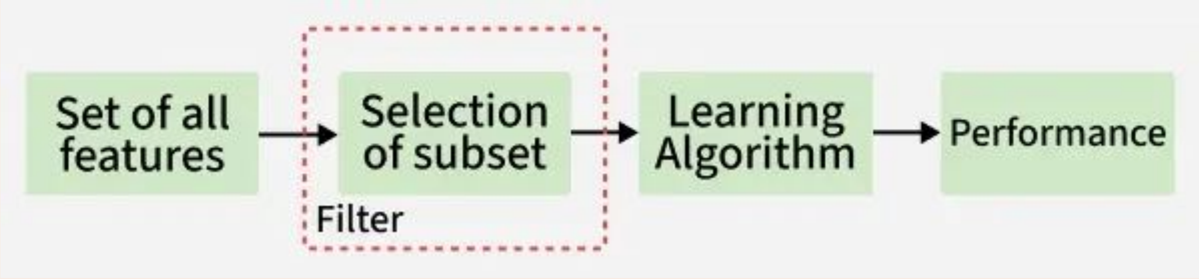
Feature selection is a core step in preparing data for machine learning where the goal is to identify and keep only the input features that contribute most to accurate predictions. By focusing on the most relevant variables, feature selection helps build models that are simpler, faster, less prone to overfitting and easier to interpret especially when we use datasets containing many features, some of which may be irrelevant or redundant.



Filter method:

Filter methods evaluate each feature independently with target variable. Feature with high correlation with target variable are selected as it means this feature has some relation and can help us in making predictions. These methods are used in the preprocessing phase to remove irrelevant or redundant features based on statistical tests (correlation) or other criteria.


Filter Method
Selection
1
2
3
4
5
Ranking
1 Rank features by statistical score
2 Select top-ranked features
3 Feed into model



Techniques for Filter method:

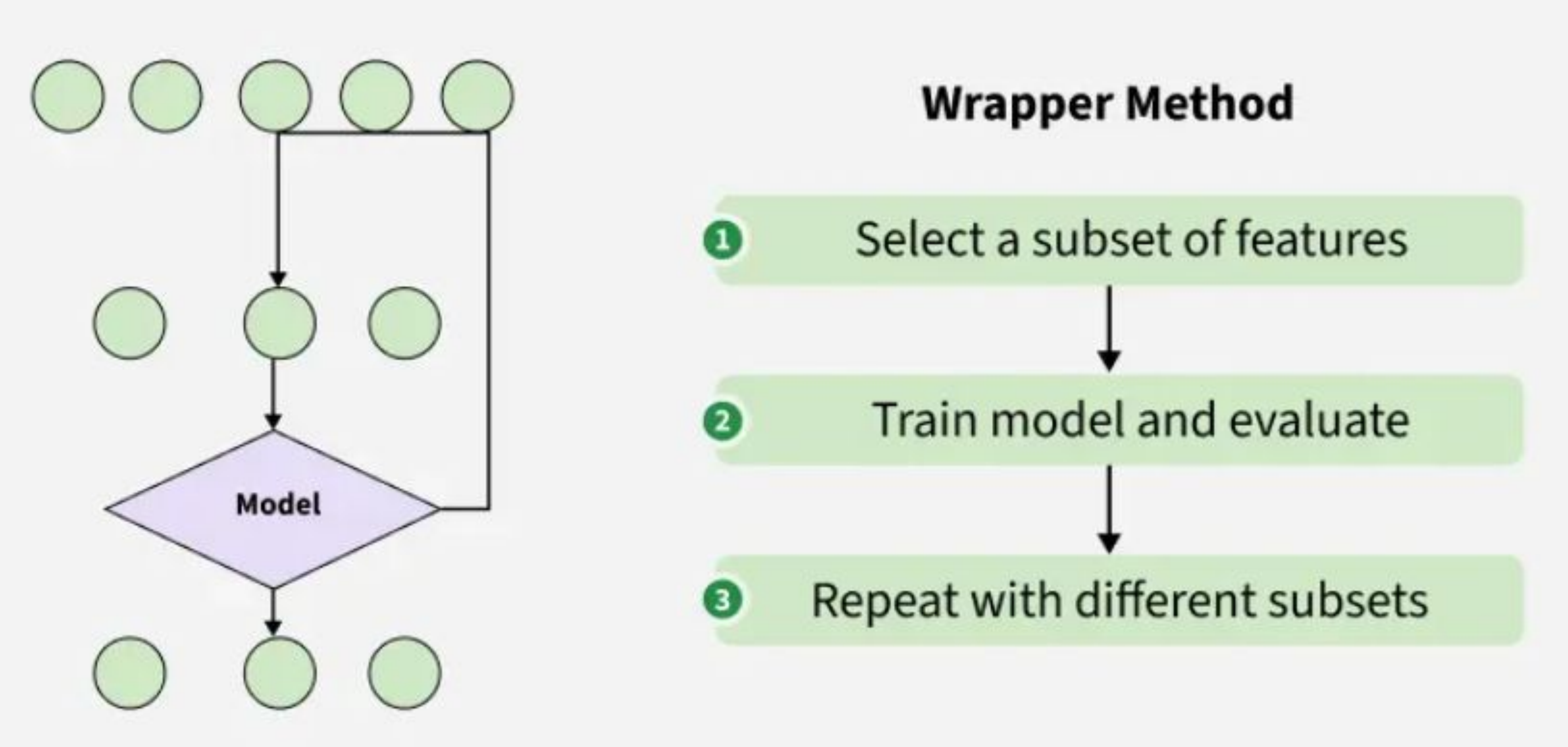
1. **Information Gain:** It is defined as the amount of information provided by the feature for identifying the target value and measures reduction in the entropy values. Information gain of each attribute is calculated considering the target values for feature selection.
2. **Chi-square test:** It is generally used to test the relationship between categorical variables. It compares the observed values from different attributes of the dataset to its expected value.
3. **Fisher’s Score:** It selects each feature independently according to their scores under Fisher criterion leading to a suboptimal set of features. Larger the Fisher’s score means selected feature is better to choose.
4. **Pearson’s Correlation Coefficient:** It is a measure of quantifying the association between the two continuous variables and the direction of the relationship with its values ranging from -1 to 1.

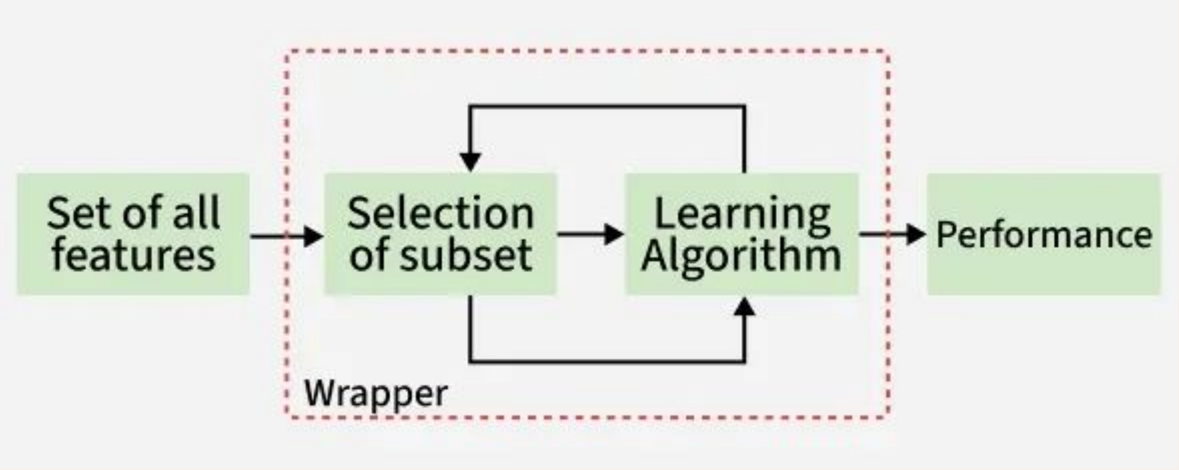
Techniques for Filter method cont.

5. **Variance Threshold:** It is an approach where all features are removed whose variance doesn’t meet the specific threshold. By default this method removes features having zero variance. The assumption made using this method is higher variance features are likely to contain more information.
6. **Mean Absolute Difference:** It is a method is similar to variance threshold method but the difference is there is no square in this method. This method calculates the mean absolute difference from the mean value.
7. **Dispersion ratio:** It is defined as the ratio of the Arithmetic mean (AM) to that of Geometric mean (GM) for a given feature. Its value ranges from +1 to infinity as AM ≥ GM for a given feature. Higher dispersion ratio implies a more relevant feature.

Wrapper method:

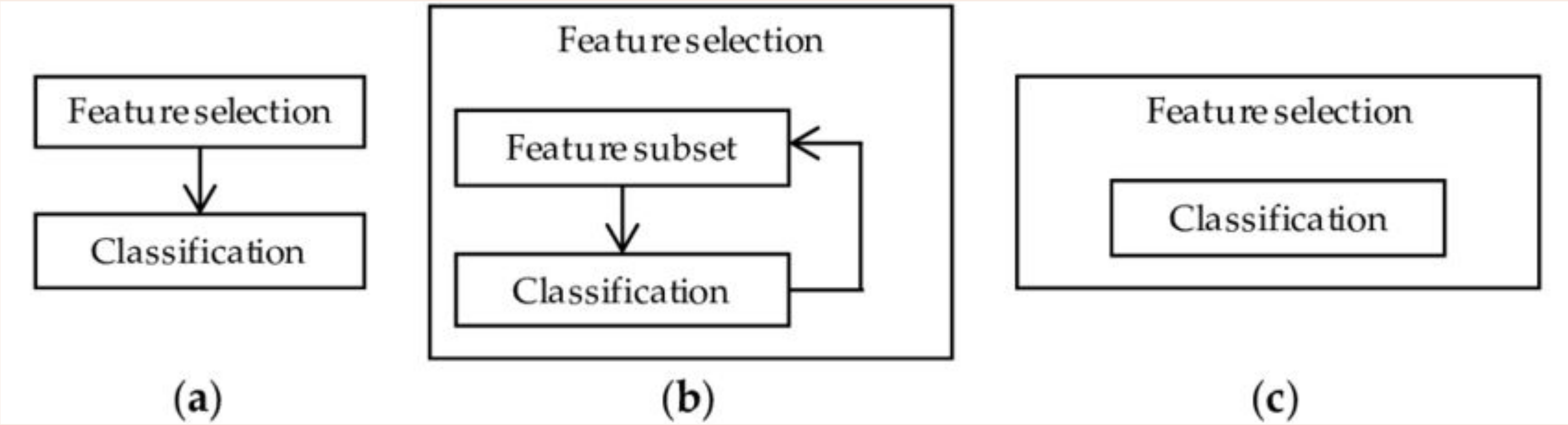
Wrapper methods are also referred as greedy algorithms that train algorithm. They use different combination of features and compute relation between these subset features and target variable and based on conclusion addition and removal of features are done. Stopping criteria for selecting the best subset are usually pre-defined by the person training the model such as when the performance of the model decreases or a specific number of features are achieved.

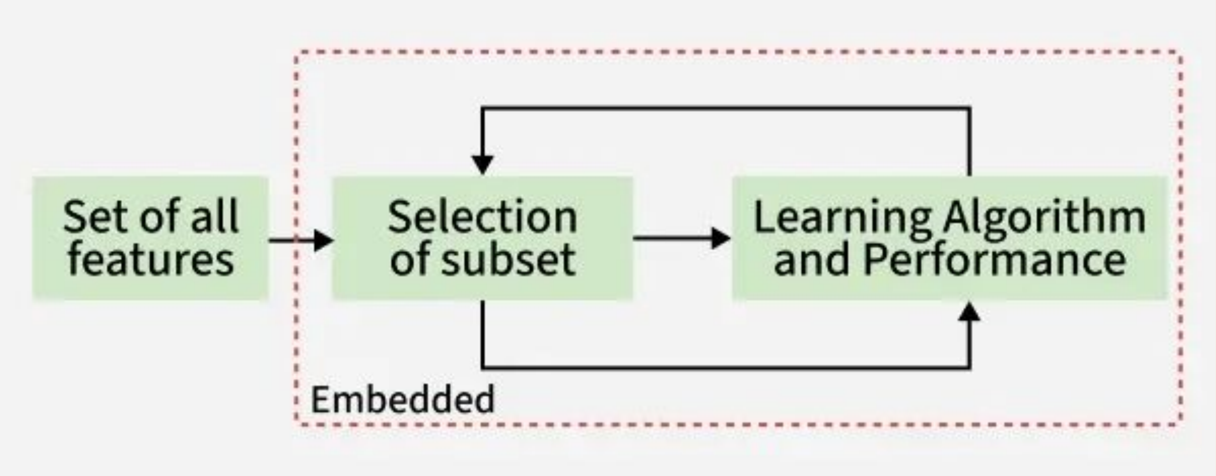

Model
Wrapper Method
1 Select a subset of features
2 Train model and evaluate
3 Repeat with different subsets



Embedded Method:

Embedded methods perform feature selection during the model training process. They combine the benefits of both filter and wrapper methods. Feature selection is integrated into the model training allowing the model to select the most relevant features based on the training process dynamically.





Choosing the Right Feature Selection Method

1. **Dataset size:** Filter methods are generally faster for large datasets while wrapper methods might be suitable for smaller datasets.
2. **Model type:** Some models like tree-based models, have built-in feature selection capabilities.
3. **Interpretability:** If understanding the rationale behind feature selection is crucial, filter methods might be a better choice.
4. **Computational resources:** Wrapper methods can be time-consuming, so consider our available computing power.