JGi JAIN DEEMED-TO-BE UNIVERSITY | Powered By Futurense

Revolutionising B.Tech

Module 5:
Feature Engineering

Course Name: **DATA EXPLORATION AND PREPARATION** [22TCSE239]

Total Hours : 09

JGi JAIN DEEMED-TO-BE UNIVERSITY
Powered By Futurense

Table of Content

- Aim
- Objectives
- Feature Selection:
  - Filter Methods
  - Wrapper Methods
  - Embedded Methods
- Feature Extraction:
  - Text Data Feature Extraction
  - Image Data Feature Extraction
  - Time Series Data Extraction
- Did You Know
- Summary
- Terminal Questions

Aim

To equip students in Analyze feature selection methods and select most appropriate for data.

Objective

A. Identify key steps in data exploration and preparation such as data cleaning, data wrangling. data transformation and data integration.

B. Ability to code in a scalable environment to automate data cleaning and processing tasks.

C. Understand the sources of data quality issues. different types of data, how data is stored and accessed.

D. Explore data visually and statistically and identify patterns and abnormalities that require further investigation.

E. Able to clean and transform data, identify missing or incorrect values, and merge or join datasets as needed.

F. Evaluate the effectiveness and efficiency of data exploration and preparation methods in terms of their impact on downstream analysis and modelling.

1. Feature Selection: Filter Methods, Wrapper Methods, Embedded Methods

1.1 Filter Methods:

- Filter methods for feature selection are a set of techniques used in machine learning to select features that are most relevant to the target variable before model training
- The primary aim is to remove irrelevant or redundant features, thus reducing the dimensionality of the dataset, which can lead to improved model performance and reduced computational costs

1.1.1 Characteristics of Filter Methods:

- **Model Independence:** They do not rely on any machine learning model to assess the importance of features.
- **Efficiency:** Generally faster and less computationally expensive than wrapper and embedded methods because they don't involve model training.
- **Univariate:** Each feature is evaluated independently based on its relationship with the target variable, without considering interactions between features.

1.1.2 Commonly Used Statistical Measures:

1. **Pearson Correlation Coefficient:** Measures the linear correlation between two continuous variables, providing insight into the strength and direction of their relationship.
2. **Chi-Squared Test:** Evaluates the independence of two categorical variables, often used for categorical features in classification problems.
3. **Mutual Information:** Assesses the mutual dependence between two variables, capturing both linear and non-linear relationships, useful for both classification and regression tasks.
4. **ANOVA (Analysis of Variance) F-test:** Determines if there are statistically significant differences between the means of three or more independent groups, often used for numerical features with a categorical target.
5. **Kendall's Tau and Spearman's Rank Correlation:** Non-parametric tests that measure the ordinal association between two measured quantities, alternative to Pearson's correlation in non-linear relationships.

1.1.3 Advantages:

Simplicity and Speed: Easy to understand and quick to execute, making them suitable for preliminary feature selection and high-dimensional datasets.

Versatility: Can be applied to any machine learning model since the selection process is independent of the model.

1.1.4 Disadvantages:

Ignores Feature Interactions: By evaluating each feature independently, filter methods may overlook interactions between features that could be predictive of the target variable when used together.

1.1.5 Application:

- Filter methods are typically used as a preliminary step in the feature selection process. They help in narrowing down the feature set to a more manageable size
- To implement filter methods in Python, libraries such as scikit-learn, pandas, and numpy are commonly used

1.2 Wrapper Methods

- Wrapper methods for feature selection involve the use of a specific machine learning model to evaluate the effectiveness of subsets of features and determine which features contribute most to the prediction accuracy of the model

1.2.1 Key Characteristics:

- **Model Dependent:** The selection of features is based on the performance of a predetermined machine learning model.
- **Search Strategy:** Utilizes a search algorithm to navigate through the space of possible feature subsets, evaluating each subset by training a model.
- **Performance Evaluation:** The main criterion for feature selection is the model's performance metric (e.g., accuracy, F1 score, etc.), computed using cross-validation or a hold-out set.

1.2.2 Common Techniques:

1. **Recursive Feature Elimination (RFE):**
   - Iteratively constructs models and removes the least important feature (or features) at each iteration based on the model weights or feature importance scores.
   - Often used with models that assign importance to features, such as linear models and decision trees.
2. **Sequential Feature Selection:**
   - **Forward Selection:** Starts with no features and adds one feature at a time, the one that most improves the model performance until no improvement is noted.
   - **Backward Elimination:** Starts with all features and removes one feature at a time, the one whose removal most improves the model performance, until no further improvement can be achieved.
   - **Bidirectional Elimination:** Combines both forward selection and backward elimination, adding and removing features to find the best subset.
3. **Genetic Algorithms:**

   Mimics the process of natural selection to select feature subsets, using operations such as mutation, crossover, and selection based on the model's performance as the fitness function.

1.2.3 Advantages:

Performance Focused: Directly aims to improve the prediction performance of the specified model.

Feature Interactions: Capable of capturing feature interactions that may be missed by filter methods.

1.2.4 Disadvantages:

Computational Cost: More computationally expensive than filter methods due to the need to train models for each feature subset considered.

Risk of Overfitting: Especially when the number of observations is not much larger than the number of features or when extensive search strategies are employed without proper validation.

```
import numpy as np
import pandas as pd
from sklearn.datasets import make_regression

# Create a synthetic dataset
X, y = make_regression(n_samples=100, n_features=5,
n_informative=3, noise=0.2, random_state=42)

# Convert to a DataFrame for easier handling
feature_names = ['study_hours', 'previous_grades',
'class_participation', 'homework_submissions',
'extra_curricular']
df = pd.DataFrame(X, columns=feature_names)
df['final_grade'] = y

# Display the first few rows of the dataset
print(df.head())
```

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

# Setup the feature matrix and target vector
X = df.drop('final_grade', axis=1)
y = df['final_grade']

# Initialize the estimator
estimator = LinearRegression()

# Initialize RFE with the estimator and number of features to
select
selector = RFE(estimator, n_features_to_select=3, step=1)

# Fit RFE
selector = selector.fit(X, y)

# Display selected features
selected_features = pd.Series(selector.support_,
index=feature_names)
print("Selected Features:\n", selected_features)
```

1.3 Filter Methods:

- Embedded methods for feature selection integrate the feature selection process within the model training algorithm.
- Unlike filter and wrapper methods, which are separate from the model learning process, embedded methods select the most important features during the model training itself.
- This approach combines the advantages of both filter and wrapper methods by considering the interaction between features while being computationally more efficient than wrapper methods.

1.3.1 Key Characteristics:

- **Integration with Model Training:** Feature selection is a part of the learning algorithm, making the process more efficient.
- **Automatic Feature Selection:** The algorithm automatically selects features as part of the model fitting process, reducing the need for separate feature selection steps.
- **Consideration of Feature Interactions:** Embedded methods can account for interactions between features directly in the feature selection process.

1.3.2 Common Techniques and Models:

1. **Lasso Regression (L1 Regularization):**
   Adds a penalty equivalent to the absolute value of the magnitude of coefficients. This can shrink some coefficients to zero, effectively performing feature selection by keeping only the non-zero coefficients.
2. **Ridge Regression (L2 Regularization):**
   While not directly used for feature selection since it doesn't set coefficients to zero, it's often mentioned in the context of regularization and feature importance.
3. **Elastic Net:**
   Combines penalties from both Lasso and Ridge. It can shrink some coefficients to zero like Lasso, making it useful for feature selection, especially when dealing with correlated features.
4. **Decision Trees and Random Forests:**
   These algorithms inherently perform feature selection by selecting the most informative features for splitting the nodes while building the tree.
5. **Gradient Boosting Machines (GBM):**
   Similar to decision trees, GBMs also select informative features for making splits but in a sequential manner, focusing on errors of the previous trees.

1.3.3 Advantages:

Efficiency: By embedding feature selection within the model training process, these methods can be more computationally efficient than wrapper methods.

Effectiveness: They can capture feature interactions and non-linear relationships directly, potentially leading to better model performance.

Simplicity: Reduces the need for a separate feature selection step, simplifying the modeling pipeline.

1.3.4 Disadvantages:

Model Specific: The selected features are tailored to the specific model used for feature selection, which may not be optimal for other types of models.

```
from sklearn.linear_model import LassoCV
from sklearn.datasets import make_regression
# LassoCV automatically performs cross-validation to find the best alpha (regularization strength)
lasso = LassoCV(cv=5).fit(X, y)

# Identify selected features (non-zero coefficients)
selected_features = lasso.coef_ != 0
print(selected_features)
print(f"Selected features: {sum(selected_features)} out of {X.shape[1]}")
```

2. Feature Extraction

2.1 Text Data Feature Extraction

- It involves transforming the text into a numerical format that machine learning algorithms can work with, essentially converting the raw text into a set of features
- Extracting features from text data in Python typically involves using libraries such as NLTK (Natural Language Toolkit), spaCy, scikit-learn, Gensim, and others

1. **Bag of Words (BoW):** It involves creating a vocabulary of all the unique words in dataset and then converting each text document into a vector where each element represents the frequency of a particular word in the document.
2. **TF-IDF (Term Frequency-Inverse Document Frequency):** This method refines the Bag of Words approach by taking into account not just the frequency of words in a single document (term frequency) but also how unique the words are across all documents (inverse document frequency). This helps in giving more weight to words that are important and unique to a document.
3. **Word Embeddings**: Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation. They are learned from the text data and can capture context, semantic relationships, and syntax.

Popular models for creating word embeddings include Word2Vec, GloVe, and FastText.

4. **N-grams:** An n-gram model considers a sequence of 'n' items from a given sample of text or speech. This method can capture the context and ordering of words, improving the model's understanding of the language. For example, bigrams (2-grams) are pairs of consecutive words, and trigrams (3-grams) are triples of consecutive words.
5. **Part-of-Speech Tagging:** This involves identifying the part of speech for each word in the corpus (nouns, verbs, adjectives, etc.) and can be used as features for machine learning models. This can help in understanding the grammatical structure of sentences.
6. **Named Entity Recognition (NER):** NER involves identifying and classifying key information (entities) in the text into predefined categories such as the names of people, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. This can be particularly useful for extracting information from text for use in databases or spreadsheets.
7. **Sentiment Analysis:** This involves extracting subjective information usually related to opinions, emotions, and sentiments. This can be used as a feature to understand the sentiment expressed in the text, which is particularly useful in analyzing customer feedback, social media, and reviews.

2.2 Image Data Feature Extraction

Feature extraction from image data involves transforming images into a format that machine learning algorithms can understand and process. This usually means converting images into a set of numerical features representing various aspects of the image, such as color, texture, shapes, and keypoints. Here are several common approaches for feature extraction from image data:

1. Color Histograms

Color histograms represent the distribution of colors in an image by counting the number of pixels that have colors in each of a fixed list of color ranges that span the color space of the image. This can be useful for image retrieval, background subtraction, and texture comparison.

2. Edge Detection

Edge detection algorithms like Canny, Sobel, and Prewitt identify the boundaries of objects within an image by detecting discontinuities in brightness. These edges can be used as features for object detection and image recognition tasks.

3. Texture Features

Texture features measure the spatial distribution of color or intensities in an image, often using methods like Gray-Level Co-occurrence Matrix (GLCM) or Local Binary Patterns (LBP). These features can help in distinguishing between different materials and surfaces in an image.

4. HOG (Histogram of Oriented Gradients)

HOG features are used to capture the structure or shape of objects in an image by quantifying the distribution of directions (orientations) of gradients (edges). HOG is particularly useful for object detection in computer vision, such as detecting pedestrians in images.

2.3 Time Series Data Extraction

The objective is to capture the underlying patterns of the time series in features that can improve the performance of machine learning algorithms

1. Statistical Features

Statistical features summarize the distribution and variability of data within a window. Common statistical features include:

Mean: The average value in a window.

Standard Deviation: Measures the dispersion of data from the mean.

Variance: The square of the standard deviation.

Skewness: Indicates asymmetry or bias of the data distribution from the mean.

Kurtosis: Measures the 'tailedness' of the distribution.

Quantiles: Including median, quartiles, and percentiles, describe the distribution's spread.

2. Temporal Features

Temporal features capture information related to time:

Trend components: Can be extracted using regression models or decomposition methods to capture upward or downward movements over time.

Seasonality components: Identify regular patterns within fixed time intervals, such as daily, weekly, or monthly cycles.

Day part features: Time of day or day of the week can impact the time series behavior.

3. Differencing Features

Differencing is a method to make the time series stationary:

First Difference: The difference between consecutive observations.

Seasonal Difference: The difference between an observation and a previous observation from the same season or cycle.

4. Rolling Window Features

Rolling or moving window features capture the local behaviors of the time series:

Rolling Mean: The average over a sliding window.

Rolling Variance or Standard Deviation: Measure of dispersion over a sliding window.

Window-based Quantiles: Percentiles calculated within a window.

Tools and Libraries for Feature Extraction

Python's Pandas library is excellent for handling time series data and simple statistical features.

NumPy can be used for more complex numerical operations.

Statsmodels provides tools for autocorrelation and model-based features.

Scikit-learn for rolling window transformations.

Tslearn or **TsFresh** libraries offer advanced time series feature extraction capabilities.

Self Assessments and Activities

Here are some assessments you can use to evaluate students understanding of DATA EXPLORATION AND PREPARATION.

- Quizzes
- In class Participation
- Write up assessments

Did You Know?

A statistician and data scientist known for creating several widely used R packages for data analysis and visualization

HADLEY WICKHAM

Summary

Outcomes: students able to

a. Analyze feature selection methods and select most appropriate for data..

Terminal Questions

1. Explain Filter Methods
2. Explain Wrapper Methods
3. Explain Embedded Methods
4. Explain Text Data Feature Extraction
5. Explain Image Data Feature Extraction
6. Explain Time Series Data Extraction.

Reference Links

TEXTBOOKS:

- Python for Data Analysis, Data Wrangling with Numpy, Pandas and Jupyter, Third Edition, O’Reilly.
- Hands on Exploratory Data Analysis using Python, Packt.

REFERENCES:

- Peter Bruce, Andrew Bruce and Peter Gedeck, Practical Statistics for Data Scientist, O’Reilly.
- Avinash Navlani, Armando Fandango and Ivan Idris, Python Data Analysis, Third Edition, Packt.

Thank you