Python 資料分析與機器學習應用

Data Analysis and Machine Learning with Python

Midterm Test

B09901061 電機四 莊政達

I. 1. T 2. F 3. T 4. T 5. T
II. 1. a 2. b 3. d 4. a 5. d
III. In python code (q1.py, q2.py, q3.py)
IV.

1. Data cleaning's importance stems from the fact that the quality and correctness of the data analysis results directly depend on the quality of the data. The following are some advantages of data cleaning.
    a. Improves Accuracy: Cleaning data is crucial for accuracy in data analysis and machine learning models. Even the most sophisticated algorithms can produce erroneous outcomes if the input data is inaccurate or inconsistent.
    b. Enhances Efficiency: Cleaning data helps in removing redundancies and irrelevant information, making data processing more efficient and reducing computational costs.
    c. Increases Productivity: Data scientists and analysts spend a significant portion of their time cleaning and preparing data. Automating common data cleaning tasks can significantly increase their productivity.

    Below are some common Data Cleaning Techniques:
    a. Handling Missing Values
    Filling missing values with the mean, median, or mode of the column, or removing rows/columns with a significant number of missing values.
    b. Removing Duplicates
    Identifying and removing duplicate entries to ensure each transaction is only counted once for accurate sales analysis.
    c. Correcting Inconsistencies & Standardizing Data
    Standardize data to a single format, like time, date, and age.

2. Significance of Overfitting:
    a. Reduces Model Generalization: Overfitting let models perform well on training data but poorly on validation or test data.
    b. Impairs Model Performance: Overfitting undermines this by making the model less reliable and accurate in real-world applications.
    c. Misguides Decision-Making: For decision-making processes that rely on predictive modeling, overfitted models lead to incorrect conclusions and potentially costly errors.

    Some Methods to Avoid Overfitting:

    a. Cross-Validation: Divides the dataset into several subsets, then trains the model on some subsets while using the remaining subsets for validation.
    b. Train with More Data: Providing more data can help the algorithm detect the signal better. However, it's also crucial to ensure the diversity and representativeness of the data.
    c. Simplify the Model: Choosing a simpler model with fewer parameters can inherently reduce the risk of overfitting. This might mean opting for linear models over complex non-linear models when appropriate. But this method isn't suitable for every scenario.
    d. Regularization: Techniques such as L1 and L2 regularization add a penalty on the size of coefficients. Regularization can discourage the model from fitting the training data too closely.

3. Comparing Decision Trees with Random Forest

Decision Trees:

- **Principle**: Splits data based on features, creating a single tree where each path to a leaf represents a classification or regression rule.
- **Advantages**: Easy to interpret; handles both numerical and categorical data.
- **Disadvantages**: Prone to overfit; unstable with small data changes; not great for continuous variables.
- **Applications**: Medical diagnosis, credit risk assessment, deciding whether to go out or not according to the weather.

Random Forests:

- **Principle**: Ensemble decision trees using bootstrap samples and feature randomness to improve prediction accuracy and reduce the chance of overfitting.
- **Advantages**: High accuracy; handles overfitting well; effective in high dimensionality spaces.
- **Disadvantages**: Hard to interpret; computationally intensive.

- **Applications**: fraud detection, e-commerce recommendations.

Summary

- **Accuracy**: Random Forests generally offer higher accuracy through ensemble learning.
- **Interpretability**: Decision Trees are simpler and easier to interpret.
- **Use Case**: Decision Trees are suited for applications requiring transparency in decision-making, while Random Forests are preferred for problems where prediction accuracy is paramount.

4. Comparing Matplotlib with Seaborn

Matplotlib

Pros:

- Versatility: Offers a wide range of plotting options and is highly customizable.
- Compatibility: Integrates well with many Pandas operations.
- Control: Provides detailed control over almost every aspect of a plot.

Cons:

- Complexity: High customization options means requiring more code than other libraries.
- Aesthetics: Default styling is considered less modern and visually appealing compared to some newer libraries like Seaborn.

Suitable Usage:

- Detailed plotting.
- When working within a broader Matplotlib ecosystem, like plotting with Pandas or customizing plots.

Seaborn

Pros:

- Ease of Use: Built on top of Matplotlib, it offers a higher-level interface for creating common statistical plots easily.
- Aesthetics: Provides more attractive default styles and color palettes to create visually appealing plots.
- Functionality: Supports the creation of complex visualizations, such as heatmaps, time series, and violin plots, with a simpler syntax.

Cons:

- Customization: While it offers simplicity and good defaults, it might be limiting when customization is needed.
- Learning: Users unfamiliar with Matplotlib might need time to understand how to extensively tweak Seaborn plots.

Suitable Usage:

- For exploratory data analysis where quick and beautiful visualizations are preferred.
- When statistical visualization capabilities are needed, such as for showing distributions, correlations, and regression models.

5.

When it comes to detecting anomalies, both random forest regression and decision tree regression are not so appropriate to be utilized.

Starting from the decision tree, since it's designed to learn decision rules from data and make proper predictions, decision trees could potentially highlight anomalies as data points that fall into very small leaves or leaves with significantly different target values compared to the majority. However, it's an indirect method and might misunderstand data in the majority as anomalies just because it's in a small leaf, or even worse, overfit to a known outlier.

Random forest regression holds similar characteristics. If a data point consistently results in significantly varied predictions across the trees, it might be considered an outlier. However, it's still indirect and not inherently suited for identifying anomalies and will still cause the model to get familiar with specific anomaly or anomalies and lead to overfitting or prediction failure to new data.