4.5 Summary

This tutorial presents Python programming examples for data preprocessing, including data cleaning (to handle missing values and remove outliers as well as duplicate data), aggregation, sampling, discretization, and dimensionality reduction using principal component analysis.

References:

1. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
2. Mangasarian, O.L. and Wolberg, W. H. (1990). "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, pp 1 & 18.
3. Wolberg, W.H. and Mangasarian, O.L. (1990). "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", Proceedings of the National Academy of Sciences, U.S.A., Volume 87, pp 9193-9196.
4. Climate Data Online [https://www.ncdc.noaa.gov/cdo-web/].

Homework 3

This homework is mainly for you to learn. You may work in a group of 2. The answer to each problem should be at least 80 words.

Your Homework 3 submission should only include content from this point on. pdf, not ipynb.

Use your own words, as using AI for your homework is not allowed and strongly discouraged.

Q-1: K-Nearest Neighbors (KNN)

Question: What are the key factors that influence the performance of the K-Nearest Neighbors (KNN) algorithm? How do you choose the value of 'K', and what is the effect of different distance metrics on classification accuracy?

there are a number of different factors affect KNN performance . The choice of distance metric is important because KNN is highly sensitive to feature magnitudes and scales, which can skew results if features aren't properly normalized before training. The value of K determines how many neighbors influence predictions. Smaller K values can lead to overfitting and noise sensitivity, while larger K values may cause underfitting. Data dimensionality also matters since KNN suffers from the curse of dimensionality in high-dimensional spaces.

We choose the optimal K value using cross-validation techniques. This involves testing different K values systematically and evaluating performance metrics like accuracy or error rates. We plot these metrics against various K values to identify the point where performance stabilizes or begins to decline, similar to finding an elbow in the curve.

Different distance metrics will also affect classification accuracy by changing how similarity between data points is measured. Euclidean distance works well for continuous features with similar scales, Manhattan distance is more robust to outliers and works better in high dimensions, while Hamming distance is specifically designed for categorical data. The choice depends on your data's distribution patterns.

Q-2: Naive Bayes

Question: Why is Naive Bayes considered a "naive" classifier, and how does it make predictions? What are the implications of the naive independence assumption in real-world applications?

Naive Bayes is considered "naive" because it makes a strong assumption that all features are conditionally independent given the class label. This means it assumes that the value of one feature has no effect on any other feature when determining the class. While this assumption is not commonly true in real-world data, it allows for simplified mathematical calculations and makes the algorithm computationally efficient.

The algorithm makes predictions by applying Bayes theorem to calculate probabilities. It multiplies the prior probability of each class by the likelihood of observing each feature value, then normalizes these products to get final probabilities. The class with the highest posterior probability becomes the predicted outcome.

The naive independence assumption has issues in real-world applications. It can decrease accuracy when features are strongly correlated, such as age and income in demographic data, or word co-occurrences in text analysis. Although, despite violating this assumption, Naive Bayes often performs surprisingly well in practice, especially for text classification and spam filtering.

Q-3: Support Vector Machines (SVM)

Question: Explain how Support Vector Machines (SVM) work in separating classes of data. What role do the kernel functions play, and how would you select a kernel function for a given dataset?

SVMs work by finding an optimal hyperplane that separates different classes of data points while maximizing the margin between them. The margin is the distance between the hyperplane and the closest data points from each class, called support vectors. This margin maximization helps ensure good generalization to unseen data by creating the most robust decision boundary possible in the feature space.

Kernel functions are important when data is not linearly separable in its original feature space. They enable the "kernel trick," which allows SVMs to implicitly map data into higher dimensional spaces where linear separation becomes possible, without explicitly calculating coordinates in this higher space. Common kernels include linear polynomial, RBF (Gaussian), and sigmoid functions

Selecting an appropriate kernel function depends on data characteristics. Use linear kernels for linearly separable data or when you have many features relative to samples. RBF kernels work well for complex, non-linear patterns and smaller datasets. Consider computational cost, as linear kernels are faster for large datasets.

Q-4: Logistic Regression

Question: How does logistic regression differ from linear regression, and why is it better suited for binary classification tasks? Explain the concept of maximum likelihood estimation (MLE) in logistic regression.

Logistic regression differs from linear regression in a feww key ways. While linear regression predicts continuous values that can range from negative to positive infinity, logistic regression uses a sigmoid function to constrain outputs between 0 and 1, making it useful for probability estimation. Linear regression assumes a linear relationship between features and the target variable, while logistic regression models the log-odds of the probability.

Logistic regression is better suited for binary classification because it naturally outputs probabilities that can be directly interpreted as the likelihood of belonging to a particular class. The sigmoid function ensures these probabilities are meaningful and bounded, and they can be easily converted to binary predictions using a threshold value typically 0.5. Maximum likelihood estimation (MLE) in logistic regression works by finding parameter values that maximize the likelihood of observing the given training data.

Unlike linear regression which uses least squares optimization, logistic regression uses MLE because we're dealing with probabilities rather than continuous values. The algorithm iteratively adjusts the coefficients using optimization techniques like gradient descent to maximize the probability of correctly predicting the observed outcomes in the training dataset.

Q-5: Data Preprocessing

This question is related to data precessing techniques introduced in this notebook. Summarize something new you learned from this noteook.

After exploring the notebook , I learned about discretization, which is the process of converting continuous numerical values into discrete categories or bins. While I had known about this technique conceptually, seeing its practical implementation was particularly valuable. The notebook demonstrated how discretization can help reduce the impact of outliers and make continuous variables more suitable for certain algorithms that work better with categorical data.

Additionally, it was interesting to get a comprehensive refresher on dealing with outliers by computing z-scores and visualizing these values within our larger dataset. The visual approach to identifying outliers through various plotting techniques helped me understand how statistical measures translate to real patterns in the data, making outlier detection much more intuitive and interpretable for practical applications.