Lecture 4 - Exploratory Data Analysis [using Python]

Alon B. Muhame

October 10, 2024

This lecture note will try to discuss what it means to get started with exploring the data in data science and machine learning:

1. Descriptive data analysis
2. Data cleansing
3. Visualization of the data during exploratory data analysis

Exploratory data analysis (EDA)

So far, our discussion of data science and machine learning concepts has focused on theoretical concepts that lend itself to application of concepts to the practice of fitting data science and machine learning models. The theory is important to make our life very easy when actually doing the practice leveraging Python environment, but we have to try and understand some initial steps on the starting the process of actually doing data science and/or machine learning. We'll consider primary cases. First, for explaratory data analysis for continuous DataFrame. Then, we'll consider data cleansing while conducting EDA. Finally, data visualization during EDA.

EDA is an approach to analyzing data sets to summarize their main characteristics (descriptive patterns and trends), often with visual methods. The goal of EDA is to uncover underlying patterns, detect anomalies, and identify relationships between variables, all of which can guide further investigation or modeling. EDA is typically conducted early in the data analysis process to understand the nature of the data before applying more formal statistical methods or building predictive models

Key objectives of EDA

- **Data Summarization**: Understand the basic properties and structure of the data, including summary statistics such as mean, median, variance, range, and distribution characteristics.
- **Identify Patterns and Relationships**: Use visual methods (e.g., scatter plots, histograms, correlation matrices) to identify patterns, trends, or relationships between variables.

- **Detect Anomalies and Outliers**: Visualize the distribution of data points to identify potential outliers or anomalies that may require further investigation.
- **Formulate Hypotheses**: Generate and test hypotheses about the underlying processes that generated the data.
- **Data Cleaning and Preprocessing**: Identify missing values, inconsistencies, or errors in the data that need to be addressed before further analysis.

Common Techniques and Tools in EDA

1. **Summary Statistics**: Compute basic statistics (e.g., mean, median, standard deviation, quartiles) to describe the central tendency and dispersion of the data.
2. **Univariate Analysis**: Analyse individual variables one at a time to understand their distributions and characteristics (e.g., histograms, box plots).
3. **Bivariate Analysis**: Explore relationships between pairs of variables using scatter plots, correlation matrices, or cross-tabulations.
4. **Multivariate Analysis**: Examine interactions between multiple variables simultaneously, potentially using techniques like dimensionality reduction (e.g., PCA) or clustering.
5. **Visualization**: Use graphical techniques (e.g., scatter plots, bar charts, heatmaps) to visually represent the data and identify patterns or anomalies.
6. **Handling Missing Data**: Investigate the presence and patterns of missing values and decide on appropriate strategies for imputation or removal.

Example 1

Steps in Exploratory Data Analysis

***Data Collection**: Gather the relevant data from various sources and ensure it is properly formatted for analysis.*

2. ***Data Cleaning**: Address missing values, outliers, and inconsistencies in the data.*
3. ***Univariate Analysis**: Explore each variable individually to understand its distribution and summary statistics.*
4. ***Bivariate Analysis**: Examine relationships between pairs of variables to identify potential associations.*
5. ***Multivariate Analysis**: Explore interactions and dependencies among multiple variables to gain deeper insights.*
6. ***Visualization**: Use visualizations to represent data patterns and relationships effectively.*
7. ***Hypothesis Testing**: Formulate and test hypotheses based on observed patterns or relationships.*

EDA visualization

Exploratory Data Analysis (EDA) involves using various visualization techniques to gain insights into the dataset's characteristics, relationships between variables, and underlying patterns. Visualization plays a crucial role in EDA as it enables analysts to explore data intuitively and identify interesting features that may guide subsequent analysis or modeling tasks. Below are some common visualization techniques used in EDA:

Comment 1

1. ***Univariate Visualization***:
   Histograms. *The purpose: Visualize the distribution of a single numerical variable. Sample code using pyplot library: plt.hist(data['column'], bins=20)*
   Box Plots: *The purpose: Display the summary statistics (median, quartiles, outliers) of a numerical variable. sample code using seaborn library: sns.boxplot(x =' category_column', y =' numeric_column', data = data)*
   Kernel Density Plots. *The purpose: Estimate the probability density function of a numerical variable. sample code usinf sea born library : sns.kdeplot(data['numeric_column'], shade = True)*

2. ***Bivariate Visualization***:
   Scatter Plots. *The purpose: Explore the relationship between two numerical variables. sample code using pyplot library: plt.scatter(data['x_column'], data['y_column'])*
   Pair Plots. *The purpose: Visualize pairwise relationships between multiple numerical variables. sample code using sea born library : sns.pairplot(data)*
   Bar Charts. *the purpose: Compare categorical variables or summarize numerical data across categories. sample code using sea born library: sns.barplot(x =' category_column', y =' numeric_column', data = data)*

3. ***Multivariate Visualization Heatmaps***. *the purpose: Display correlation matrices or relationships between multiple variables. sample code using sea born library: sns.heatmap(data.corr(), annot=True, cmap='coolwarm')*
   3D Scatter Plots. *The purpose: Visualize three-dimensional relationships between variables. sample code xxample: frommpl_toolkits.mplot3dimportAxes3Dfig = plt.figure()ax = fig.add_subplot(111, projection =' 3d')ax.scatter(data['x_column'], data['y_column'], data['z_column'])*
   Parallel Coordinates. *The purpose: Visualize high-dimensional data by plotting multiple variables along parallel axes. sample code example: frompandas.plottingimportparallel_coordinatesparallel_coordinates(data[['var1',' var2',' var3',' var4',' class']],' class')*

4. ***Time Series Visualization Line Plots***. *The purpose: Display trends and patterns in time series data. sample example: data['date_column'] = pd.to_datetime(data['date_column'])sns.lineplot(x =' date_column', y =' value_column', data = data)*
   Seasonal Decomposition. *The purpose: Decompose time series data into trend, seasonal, and residual components. Example: fromstatsmodels.tsa.seasonalimportseasonal_decomposeresult = seasonal_decompose(data['value_column'], model =' additive')result.plot()*

Comment 2

1. ***Interactive Visualization Plotly and Bokeh python plotting libraries***. *The purpose: Create interactive visualizations for exploring complex datasets. Example:* $importplotly.expressaspxfig = px.scatter(data, x =' x_column', y =' y_column', color =' category_column', hover_data = ['additional_info'])fig.show()$ *Additional Tips* ***Customization****: Customize plots using parameters like colors, labels, titles, and annotations to enhance readability and clarity.* ***Multiple Plots****: Combine multiple plots using subplots to compare different aspects of the data.* ***Iterative Exploration****: Explore data iteratively, focusing on different variables or relationships based on initial observations.*

Data cleansing

Data cleansing, also known as data cleaning or data preprocessing, is a crucial step in machine learning (ML) where the raw data collected for a particular problem is transformed, formatted, and cleaned to prepare it for model training. The goal of data cleansing is to improve the quality and reliability of the data by addressing issues such as missing values, noise, duplicates, and inconsistencies. Clean data is essential for ML models to learn effectively and produce accurate predictions.
Handling Missing Value. The first step in handling missing values is Identification. That is identify columns or features with missing values. In addition, understand the Cause of missingness. That is determine why values are missing (e.g., data not recorded, technical issues).
Strategies for Handling:

1. Removal (Deleting Missing Data). Remove rows or columns with a high percentage of missing values.

   Handling missing values in data is a crucial aspect of data preprocessing in machine learning. Different techniques such as removal, imputation, and advanced methods can be used to address missing data, each with its own set of advantages and limitations. Let's examine the pros and cons of each approach:
   Pros:

   - Simplicity: It is straightforward and easy to implement.
   - Preservation of Data Quality: By removing rows or columns with missing values, the integrity and quality of the remaining dataset are maintained.

- Useful for Large Datasets: In large datasets, removing a small percentage of missing data may not significantly impact the overall analysis.
  Cons:
- Loss of Information: Removing missing data can lead to loss of potentially valuable information, especially if missing values are not randomly distributed.
- Reduction in Sample Size: Removing rows with missing values reduces the sample size, which can affect the representativeness of the dataset.
- Bias in Analysis: If missing values are not completely at random (MAR) or missing completely at random (MCAR), removal can introduce bias into the analysis.

2. Imputation Replace missing values with statistical measures like mean, median, or mode of the column.

   Pros:

   - Retains Data Volume: Imputation allows you to retain all rows and columns of the dataset.
   - Preserves Relationships: Imputation methods can help preserve relationships and patterns in the data.
   - Prevents Biases: By keeping the sample size intact, imputation can reduce bias that might arise from removing missing values.
   - Cons:
   - Introduction of Inaccuracies: Imputed values may not accurately represent the true values and can introduce noise or bias into the data.
   - Impact on Variability: Imputation can reduce the variability of the dataset, potentially affecting the performance of models.
   - Method Selection: The choice of imputation method can influence results, and the optimal method may vary depending on the dataset and context.

3. Advanced Techniques: Use predictive models to estimate missing values based on other features. **Pros:**

   - Preserves Data Quality: Advanced techniques such as predictive modelling for imputation can provide more accurate estimates of missing values.
   - Utilizes Relationships: These methods leverage relationships within the dataset to make informed imputation decisions.

- Can Improve Model Performance: Predictive imputation may lead to improved performance in downstream modelling tasks by retaining valuable information.
  Cons:
- Complexity: Advanced techniques require more computational resources and may be more complex to implement.
- Overfitting: Predictive imputation methods can potentially overfit the imputation model to the observed data, leading to biased results.
- Sensitive to Model Choice: The effectiveness of predictive imputation depends on the choice of modelling approach and parameters.

4. Choosing the Right Approach
   - Nature of Missingness: Understanding the nature and pattern of missing data (MCAR, MAR, or MNAR) can guide the choice of technique.
   - Data Size and Quality: Consider the size of the dataset, the impact of missing values, and the overall quality of the data before deciding on a method.
   - Domain Knowledge: Domain-specific knowledge can help in selecting the most appropriate technique based on the characteristics of the data and the problem at hand.
5. Outliers Outliers are data points that significantly differ from other observations in a dataset. They can occur due to various reasons such as measurement errors, experimental anomalies, or natural variations in data. Outliers can adversely affect the performance and accuracy of machine learning models by skewing the results and influencing statistical estimates. Handling outliers in machine learning involves several techniques aimed at identifying, analysing, and appropriately addressing these data points.