Lecture Note 3: Exploratory Data Analysis

1. Introduction to Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a critical step in the realm of business analytics. It involves the systematic process of examining and visualizing data sets to gain valuable insights, uncover hidden patterns, and identify initial trends. The primary objective of EDA is to understand the underlying structure of the data, which in turn aids in making informed business decisions.

During EDA, analysts examine various aspects of the data, such as its distribution, central tendencies, variability, and relationships between variables. The process often involves generating summary statistics, creating visualizations (such as histograms, scatter plots, and box plots), and conducting basic statistical tests to detect outliers or anomalies.

EDA serves several purposes in business analytics:

1. **Data Quality Assessment:** Exploratory analysis helps identify missing values, data errors, or inconsistencies. Cleaning and preparing the data for analysis is a crucial step, and EDA helps ensure the data's accuracy and reliability.
2. **Pattern Discovery:** EDA assists in identifying initial patterns and relationships within the data. For instance, it can help uncover correlations between variables, which might offer insights into customer behavior, market trends, or operational efficiencies.
3. **Feature Selection:** In predictive modeling, selecting the right features (variables) that have a significant impact on the outcome is crucial. EDA aids in identifying relevant features and understanding their potential impact on the analysis.
4. **Hypothesis Generation:** Exploratory analysis often prompts the formulation of hypotheses that can guide subsequent analysis. These hypotheses can be tested using more advanced statistical methods.
5. **Problem Refinement:** EDA can help refine the scope and objectives of the analytics project. As analysts gain a deeper understanding of the data, they may discover new angles to explore or additional questions to address.
6. **Effective Communication:** Visualizations generated during EDA provide a clear and concise way to communicate findings to stakeholders, allowing them to grasp insights quickly.

In essence, exploratory data analysis provides the foundation for subsequent steps in the analytics process. It helps analysts make informed decisions about data preprocessing, feature engineering, model selection, and more. By understanding the data's characteristics and relationships, analysts can ensure that the subsequent analyses and interpretations are based on a solid understanding of the data's nuances.

1.1 Data Cleaning and Preparation

Data cleaning and preparation involves identifying and rectifying errors, inconsistencies, and inaccuracies in the dataset. This process includes handling missing values, correcting typos, resolving duplicate entries, and addressing any anomalies that might affect the quality and reliability of the data. The primary goal of data cleaning is to ensure that the dataset is accurate, complete, and ready for analysis.

In the previous note, I proposed that steps 3 and 4 within an analytics project involve data cleaning and exploratory data analysis (EDA). It's quite common for these two steps to overlap. In fact, it's often during EDA that we come across typos, mistakes, and irregularities.

An important task of data cleaning is to address the issue of missing values. Indeed, within the customer churn dataset, numerous variables include 'unknown' as a value, which can essentially be treated as missing values. While one straightforward approach is to eliminate entire rows or columns with missing values, this method can significantly influence the performance of your models, particularly when dealing with extensive missing data or situations where the rows/columns containing missing values hold valuable information. A more effective approach involves imputing the missing values using information from the available data. This technique is referred to as imputation.

When encountering missing or unknown values within a categorical variable, a prevalent approach is to consider them as a distinct category. Alternatively, a common imputation method involves substituting them with the variable's mode. In the case of numerical variables, missing values can be substituted with the variable's mean, median, or mode. These techniques, known as univariate feature imputation, offer straightforward solutions.

Advanced imputation methods, termed multivariate feature imputation, encompass deducing missing values based on relationships with other attributes or columns. One such example is the K-Nearest Neighbor (KNN) technique for imputing missing values. However, multivariate imputation can present a drawback in the form of data leakage. This phenomenon transpires when information not meant to be accessible during prediction is utilized in model construction or training, leading to overly optimistic performance assessments.

1.2 EDA vs Data Processing and Feature Engineering

In my view, I categorize data modeling as the step that follows EDA within an analytics project. It encompasses data processing and feature engineering in addition to building data mining/machine learning models. I narrowly define data preprocessing to be the process of scaling numerical variables, encoding categorical variables, and handling outliers. The goal of data preprocessing is to create a well-structured and well-prepared dataset that can be used effectively in machine learning or statistical analysis.

The quality and relevance of features directly impact the performance of machine learning algorithms. Feature engineering is the process of creating new features (feature expansion) and selecting relevant variables (feature selection) that can enhance the model's ability to capture patterns, relationships, and information from the data. Effective feature engineering requires domain knowledge and a deep understanding of the data and problem context. It's an iterative process that involves experimentation, testing, and fine-tuning to identify the features that contribute the most to model performance. Well-engineered features can significantly enhance the interpretability, accuracy, and robustness of machine learning models, enabling them to make more informed and accurate predictions.

EDA plays a pivotal role in shaping the strategies for data processing and feature engineering within the realm of data analytics and machine learning. It serves as a critical precursor that informs decisions about how to manipulate and refine the dataset to better suit the demands of subsequent analysis.

2. EDA on a Single Categorical Variable

Exploratory Data Analysis (EDA) on a single categorical variable involves investigating the characteristics and patterns within that variable to gain insights and inform decision-making. This process helps to understand the distribution of categories, identify the most frequent categories, and potentially uncover relationships with other variables. Let's walk through some commonly used techniques of EDA on a categorical variable:

1. **Frequency Distribution:** Calculate the frequency (count) of each category in the categorical variable. This provides an overview of the distribution and reveals which categories are more or less common.
2. **Percentage (Relative Frequency) Distribution:** Calculate the percentage of each category in the variable. This allows you to understand the relative proportion of each category within the dataset.
3. **Bar Plot:** Create a bar plot or a histogram to visualize the frequency distribution of categories. This visual representation makes it easy to compare the counts of different categories and identify any significant disparities.
4. **Pie Chart:** Use a pie chart to display the percentage distribution of categories. While bar plots provide a better sense of the differences in counts, pie charts help visualize the proportional composition of categories.
5. **Mode:** Identify the mode of the categorical variable, which is the category with the highest frequency. This gives you an idea of the most prevalent category in the dataset.

Below are the frequency and relative frequency tables for the variables 'view' and 'grade' from the house sales dataset, and 'income_category' from the credit card customer churn dataset. In practice, frequency and relative frequency tables can be more effective than visualizations like bar plots or pie charts when dealing with many categories or sparse data.

Frequency and Relative Frequency Table for 'view':

view	Frequency	Relative Frequency (%)
0	19489	90.1726%
1	332	1.5361%
2	963	4.4557%
3	510	2.3597%
4	319	1.476%

Frequency and Relative Frequency Table for 'grade':

grade	Frequency	Relative Frequency (%)
1	1	0.0046%
3	3	0.0139%
4	29	0.1342%
5	242	1.1197%
6	2038	9.4295%
7	8981	41.5537%
8	6068	28.0757%
9	2615	12.0992%
10	1134	5.2468%
11	399	1.8461%
12	90	0.4164%
13	13	0.0601%

Frequency Distribution for 'Income_Category':

Income_Category	Frequency	Relative Frequency
$120K +	570	0.070362
$40K - $60K	1430	0.176521
$60K - $80K	1128	0.139242
$80K - $120K	1248	0.154055
Less than $40K	2851	0.351932
Unknown	874	0.107888

In summary, EDA on a categorical variable helps reveal the distribution, composition, and potential relationships of categories within the data. Visualizations and summary statistics derived from EDA provide a clearer understanding of the variable's characteristics and guide subsequent analysis and decision-making.

3. Implications of EDA on Single Categorical Variables

Exploratory Data Analysis (EDA) on one categorical variable involves a detailed examination of its distribution, frequencies, and patterns. This process helps us make informed decisions about the need to merge categories and which encoding technique to apply. Here I will address merging categories. I will delve into encoding categorical variables in the next section.

3.1 Merge Categories

EDA begins by understanding the distribution of categories within the categorical variable. If there are too many categories or if some categories have very low frequencies, it might be beneficial to consider merging them. This simplification can improve model performance, reduce noise, and make the variable more interpretable.

By visualizing the frequency distribution, you can identify categories that have similar characteristics or meanings. For example, in a "Customer Income" category, you might merge categories like "Low Income" and "Lower Middle Income" into a single "Low/Middle Income" category if their behaviors and characteristics are similar.

Once you identify categories to merge, you need to define clear criteria for combining them. This could involve consolidating categories with similar attributes, values, or meanings. It's important to ensure that the merged categories remain meaningful and informative for the analysis or modeling tasks.

3.2 Encode Categorical Variables

Encoding categorical variables is a crucial step in preparing data for machine learning models, as many algorithms require numerical input. Categorical variables are those that represent qualitative attributes, such as colors, types, or labels, and they can't be directly used in their original form by most machine learning algorithms. Therefore, encoding is necessary to convert categorical variables into a numerical format that algorithms can understand and process.

There are a few common methods for encoding categorical variables:

1. **Label Encoding:** In this method, each unique category is assigned a unique integer label. This is useful when the categorical variable has a natural order or hierarchy. However, using label encoding on non-ordinal categorical variables can introduce unintended relationships between categories. For instance, if we label encode colors as 0, 1, 2, etc., the algorithm might interpret higher values as more important.
2. **One-Hot Encoding:** One-hot encoding is used for nominal categorical variables, where there is no inherent order between categories. It creates a new binary (0 or 1) column for each category in the original variable. If a data point belongs to a certain category, its corresponding binary column is set to 1; all other binary columns are set to 0. This technique avoids introducing artificial relationships between categories, which is a common issue in label encoding. When using one-hot encoding, for a categorical variable with k distinct values, it's common to create k-1 binary variables. This approach is used to avoid multicollinearity, a situation where one variable can be predicted from the others, making it difficult to interpret the model's coefficients.
3. **Frequency Encoding:** Frequency encoding replaces categories with their corresponding frequency of occurrence in the dataset. This can be useful for categories that have a meaningful relationship with the target variable, as it captures the prevalence of each category.
4. **Target Encoding (Mean Encoding):** In target encoding, each category is replaced with the average of the target variable for that category. This method can be effective, but it also runs the risk of data leakage, especially if not handled carefully.

The choice of encoding method depends on the nature of the categorical variable, the algorithm being used, and the potential impact on the model's performance. It's essential to understand the underlying meaning of the categorical variable and the potential implications of different encoding choices.

Incorrect or careless encoding can lead to poor model performance, so it's important to carefully consider the categorical variables in your dataset and choose the appropriate encoding method that best preserves the information and relationships within the data.

4. EDA on a Single Numerical Variable

Exploratory Data Analysis (EDA) on a numerical variable involves using various techniques to gain insights, identify patterns, and understand the distribution and characteristics of the data. This process helps uncover hidden relationships and anomalies that can inform subsequent steps in data analysis and modeling.

Here's how you might perform EDA on a numerical variable:

1. **Summary Statistics:** Calculate basic summary statistics such as mean, median, standard deviation, minimum, and maximum. These statistics provide a quick overview of the central tendency, spread, and range of the data.
2. **Histogram:** Create a histogram to visualize the distribution of the numerical variable. This graph displays the frequency of different value ranges, giving you an idea of the data's shape (e.g., normal distribution, skewed, bimodal).

3. **Box Plot:** Construct a box plot to visualize the distribution's central tendency, spread, and potential outliers. The box represents the interquartile range (IQR), while the whiskers extend to show the data's range. Outliers are typically shown as individual points.
4. **Density Plot:** A density plot is a smoothed version of the histogram, providing a continuous representation of the data's distribution. It can reveal patterns that might not be immediately evident in a histogram.
5. **Quantile-Quantile (Q-Q) Plot:** A Q-Q plot compares the quantiles of your data's distribution to the quantiles of a theoretical distribution (usually a normal distribution). This helps you assess if your data follows a particular distribution.
6. **Descriptive Visualizations:** Create visualizations like line plots, time series plots, or bar plots (if the variable is discrete) to better understand how the numerical variable changes over time or across categories.
7. **Outlier Detection:** Identify potential outliers using statistical methods like the z-score or the IQR method. Outliers can significantly impact analysis and modeling results, so it's important to assess their impact.
8. **Hypothesis Testing:** Outliers or uncommon patterns can influence the foundational assumptions of our data models, thereby requiring hypothesis testing to determine, for instance, whether the empirical data conforms to a normal distribution.

By performing EDA on a numerical variable, you can uncover key characteristics of the data, assess its quality, and make informed decisions about subsequent analysis steps, such as feature engineering, modeling, and hypothesis testing. EDA plays a crucial role in ensuring that data-driven insights are accurate, reliable, and relevant to the problem you're trying to solve.

5. Implications of EDA on Single Numerical Variables

EDA provides valuable insights into the distribution, central tendencies, and variability of the numerical variable. These insights guide decisions about which analytical techniques are appropriate for further exploration. For example:

- If there are outliers, robust statistical methods or transformations might be necessary.
- If the variable shows distinct patterns, clustering algorithms or discretization could be applied.

5.1 Discretization

Discretization, also known as binning or bucketing, is a data preprocessing technique used to convert continuous numerical features into discrete categories or bins. This process simplifies the data by grouping values into specific ranges, making it easier to analyze patterns and relationships. Discretization is particularly useful when the original continuous data has a wide range and fine-grained details that may not be relevant for certain analyses or modeling techniques.

The steps involved in discretization are as follows:

1. **Choose the Number of Bins:** The first step is to determine the number of bins into which the data will be divided. This can be based on domain knowledge, the desired level of granularity, or the characteristics of the data distribution.
2. **Define Bin Boundaries:** The range of the continuous variable is divided into equal or unequal intervals, each forming a bin. The bin boundaries are determined based on the chosen number of bins and the data distribution.

3. **Assign Values to Bins:** Each data point is then assigned to the appropriate bin based on its value. Values falling within the range of a specific bin are grouped together.

Discretization offers several benefits:

1. **Simplification:** Binning reduces the complexity of data by converting continuous values into discrete categories, making it easier to understand and analyze.
2. **Noise Reduction:** Discretization can help smooth out noise in the data and eliminate minor variations that might not be relevant.
3. **Interpretability:** Discrete bins provide a clear interpretation of data patterns and relationships, which can be particularly useful for visualization and communication.

However, discretization also has limitations:

1. **Information Loss:** Binning can lead to information loss, especially when the original data had fine-grained details. This might impact the accuracy of certain analyses or models.
2. **Bin Boundaries:** The choice of bin boundaries can influence the results, so it requires careful consideration.
3. **Impact on Analysis:** Some analyses might require the use of continuous data, and discretization can alter the characteristics of the original data distribution.

Discretization is an effective technique for simplifying data and gaining insights from continuous numerical features, especially in cases where the original granularity is not essential for the analysis or modeling goals.

5.2 Feature Scaling and Transformation

Feature scaling or transformation is a preprocessing technique used on numerical variables in machine learning. Its purpose is to bring all numerical features to a similar scale, ensuring that their magnitudes do not disproportionately impact the learning algorithm's behavior. This is particularly crucial for algorithms sensitive to the scale of input features, like distance-based methods (e.g., k-nearest neighbors) or gradient descent optimization.

Common methods of feature scaling include:

1. **Standardization (Z-score):** This technique scales data to have a mean of 0 and a standard deviation of 1. It preserves the shape of the distribution and is suitable when data follows a normal distribution.
2. **Min-Max Scaling:** Data is transformed to a specific range, typically between 0 and 1. It is useful when features have different ranges and ensures all features are within a consistent range.
3. **Robust Scaling:** This method uses the median and interquartile range to scale data. It is less sensitive to outliers compared to standardization.
4. **Mean Normalization ($x_{scaled} = \frac{x - mean(x)}{max(x) - min(x)}$):** It scales the features by transforming them in such a way that the mean of the features becomes zero, and it typically scales the feature values within the range of -1 to 1.
5. **Log Transformation:** Applying a logarithm to the data can help normalize skewed distributions and stabilize variance.

The choice of scaling method depends on the characteristics of the data and the requirements of the algorithm being used. Scaling ensures that the algorithm converges faster, provides more accurate results, and is less influenced by varying feature magnitudes.