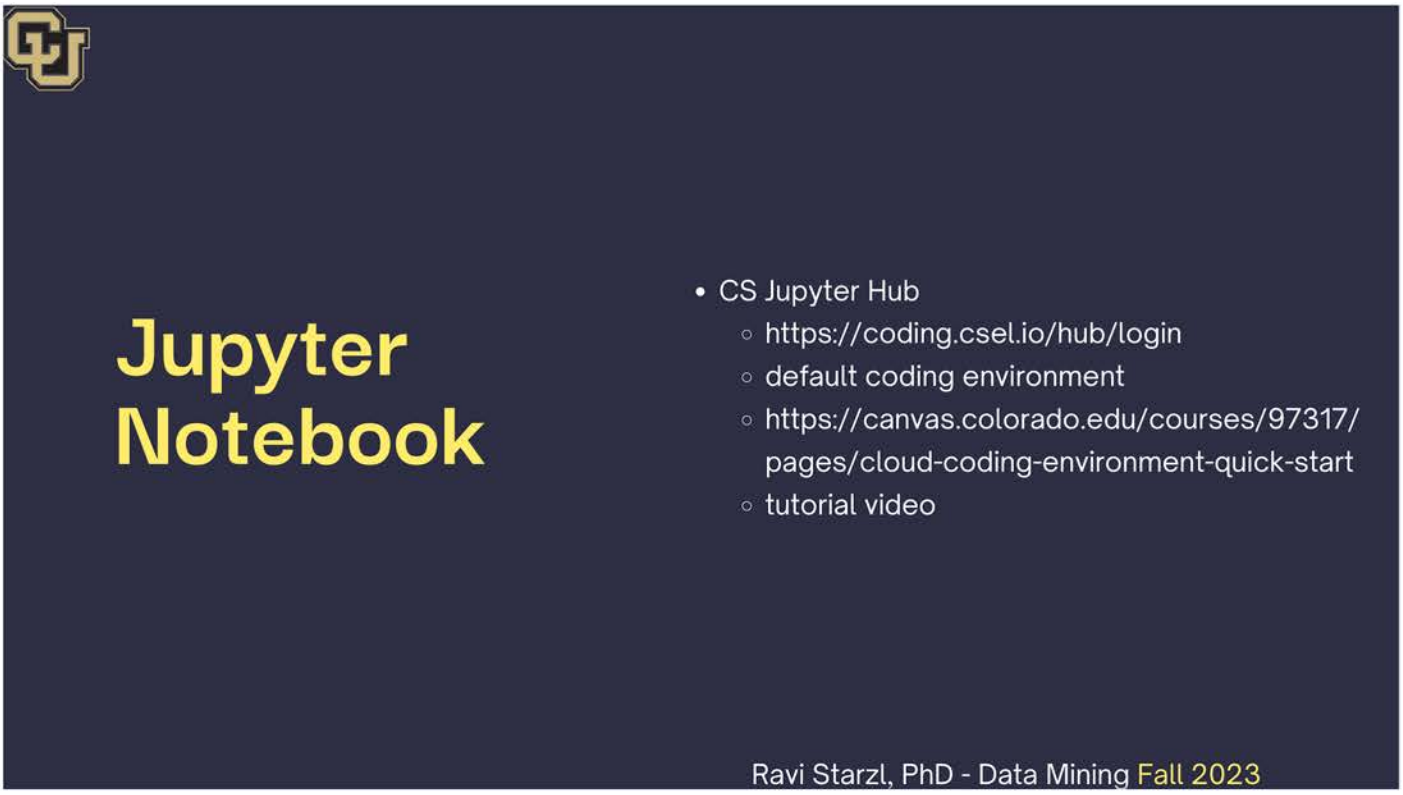
CSCI 4502/5502

Data Mining - Fall 2023 - Lecture 4

Ravi Starzl, PhD



The Jupyter Notebook is a crucial tool in the field of data science. As an open-source platform, it facilitates interactive computing, offering versatility with support for multiple programming languages, including predominantly Python. This platform enables users to integrate various elements like code, text, images, and other multimedia resources in one document, which is referred to as a notebook.

Widely employed in sectors that necessitate regular data analysis including data mining, statistical modeling, machine learning, and academic research, the Jupyter Notebook assists users in writing and running code efficiently. Users can easily observe the output, formulate hypotheses, and iterate through this process in a streamlined manner.

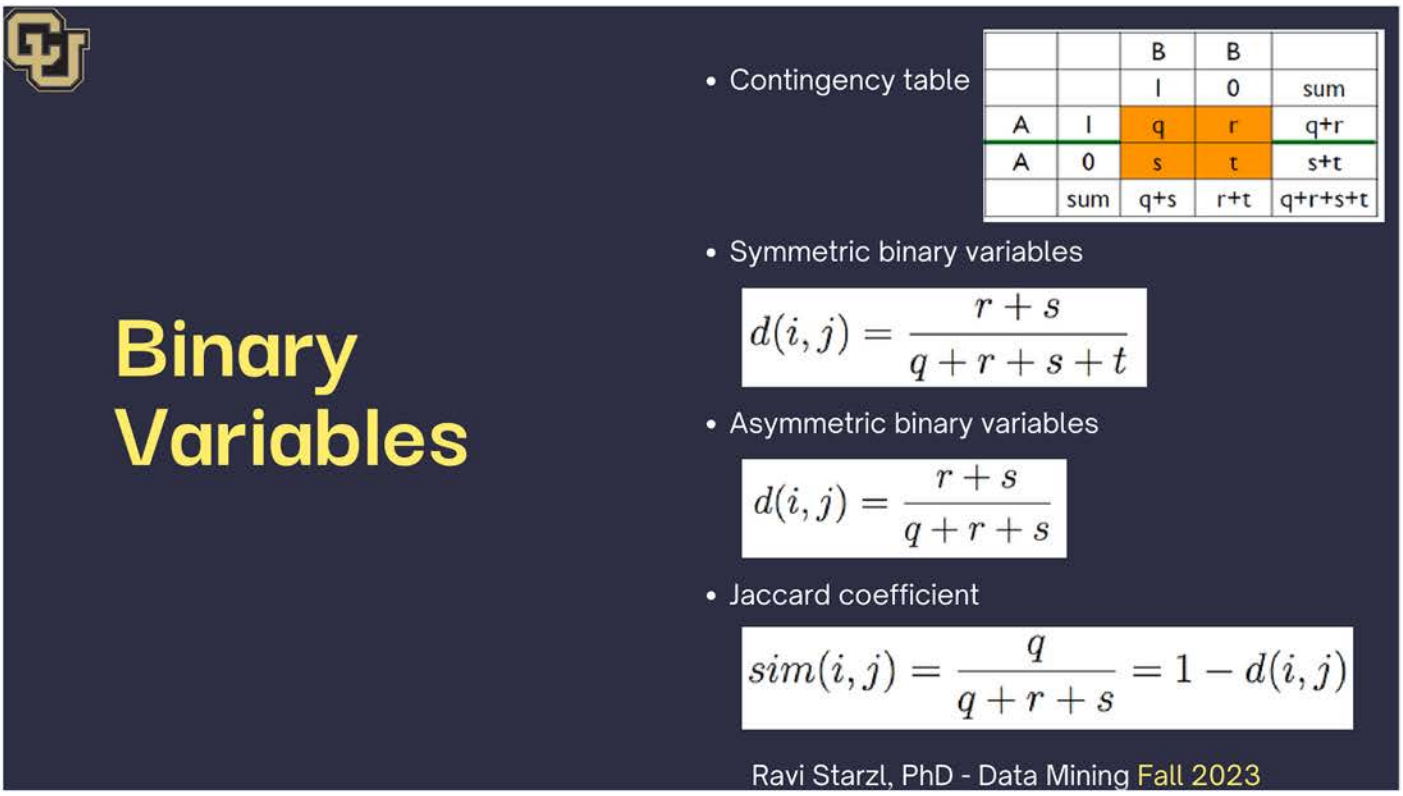
Within the context of this class, the CS Jupyter Hub serves as the main coding environment. Students can access this environment through the provided URL. This hub simplifies the coding process by allowing users to write and execute code in a web browser, negating the necessity for local software installations. This feature is especially beneficial when handling substantial datasets or tasks that require high computational power, as all the computational tasks are carried out on the server hosting the hub, not on the user's personal machine.

To assist in getting accustomed to the Jupyter environment, a tutorial video has been made available. Such videos are an effective means to guide students visually through a new software platform, potentially covering vital topics such as initiating a new notebook, executing code, and utilizing the markdown feature for creating formatted text.

A quick start guide for a cloud coding environment is accessible via the provided Canvas link. These cloud-based platforms are gaining popularity due to the ease they offer in accessing computational resources at any time and from any place, without necessitating any user setup. These platforms are particularly useful for collaborative projects, enabling multiple users to work concurrently on the same project.

Utilizing the Jupyter Notebook through the CS Jupyter Hub and cloud coding environments grants students access to adaptable, sturdy, and user-friendly platforms to explore, test, and apply data science concepts. Acquiring proficiency with these tools is highly advantageous, considering their widespread use in contemporary data science operations, and will prove beneficial.


Continuing to Get to Know Your Data
1 Data objects and attribute types
2 Basic statistical description of data
3 Data visualization
4 Measuring data similarity and dissimilarity
Ravi Starzl, PhD - Data Mining Fall 2023



Binary variables are those that can take only two possible values, which are generally represented as 0 and 1. These variables are common in computer science to indicate states such as off/on or false/true, and in statistics to depict outcomes like success/failure or presence/absenc e.

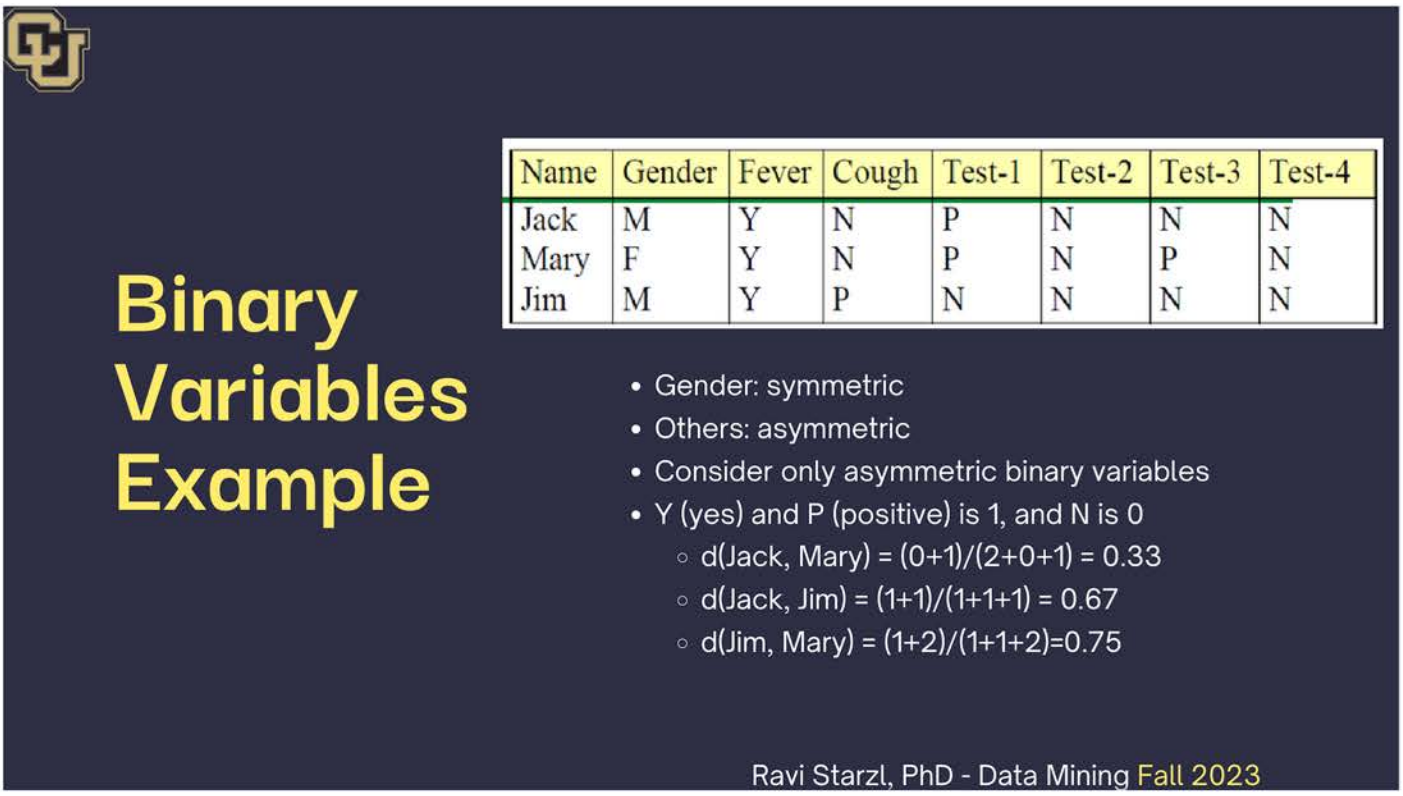
A notable concept in the analysis of binary variables is the contingency table. This matrix format table demonstrates the frequency distribution of variables. When dealing with binary variables, a 2x2 contingency table can be used to illustrate the relationship between two binary variables efficiently. For instance, in a clinical study, a contingency table might display the number of patients with positive or negative responses to a treatment compared to a placebo.

Understanding binary variables further divides them into symmetric and asymmetric categories. Symmetric binary variables regard both outcomes, 0 and 1, as equally significant. An example of this is representing an individual's gender as male (1) or female (0), with both states having equal importance. Conversely, asymmetric binary variables feature outcomes where one state holds more importance than the other. A typical example would be a medical test for a disease, wherein a positive outcome (1) holds more gravity than a negative outcome (0).

In analyzing the dissimilarities between binary variables, specific formulas are used. In these formulas, 'q' refers to the instances where both objects hold a value of 1, while 'r' and 's' denote the instances where the objects have contrasting values, and 't' represents instances where both objects have a value of 0. For symmetric binary variables, the formula d(I,j) = (r+s)/(q+r+s+t) is used to calculate the proportion of attributes where the variables differ, considering both

outcomes as equally significant. In contrast, the formula for asymmetric binary variables is d(I,j) = (r+s)/(q+r+s), excluding 't' from the denominator as the zero state is less important in this case.

Introducing the Jaccard coefficient, a metric used to assess the similarity between finite sample sets. This coefficient is given by the equation J(I,j) = q/(q+r+s), which calculates the ratio of shared attributes (where both objects are 1) to all attributes where at least one object has a value of 1. A higher Jaccard coefficient indicates a greater similarity between the sets being compared.



To deepen our grasp on the concept of binary variables, let's consider a practical example that focuses on calculating the dissimilarity between objects characterized by these variables. Suppose we have data for three individuals - Jack, Jim, and Mary, represented in binary format. We can imagine this data consisting of several attributes, such as preference for a certain genre of music, pet ownership, or adherence to a vegetarian diet, with the potential responses being "Yes" (1) or "No" (0).

In this dataset, the 'Gender' attribute is treated as symmetric since both states, Male (1) and Female (0), hold equal importance. The other attributes are regarded as asymmetric, implying that a "Yes" (1) response holds more weight compared to a "No" (0) response.

Our objective is to gauge the level of dissimilarity between the individuals based on the given binary attributes. To achieve this, we apply the previously introduced formula for asymmetric binary variables: $d(I,j) = (r+s)/(q+r+s)$. In this equation, 'q' represents the count of attributes where both individuals have marked '1', 'r' and 's' indicate the instances where the responses differ, and 't' accounts for the attributes where both individuals responded with '0'.

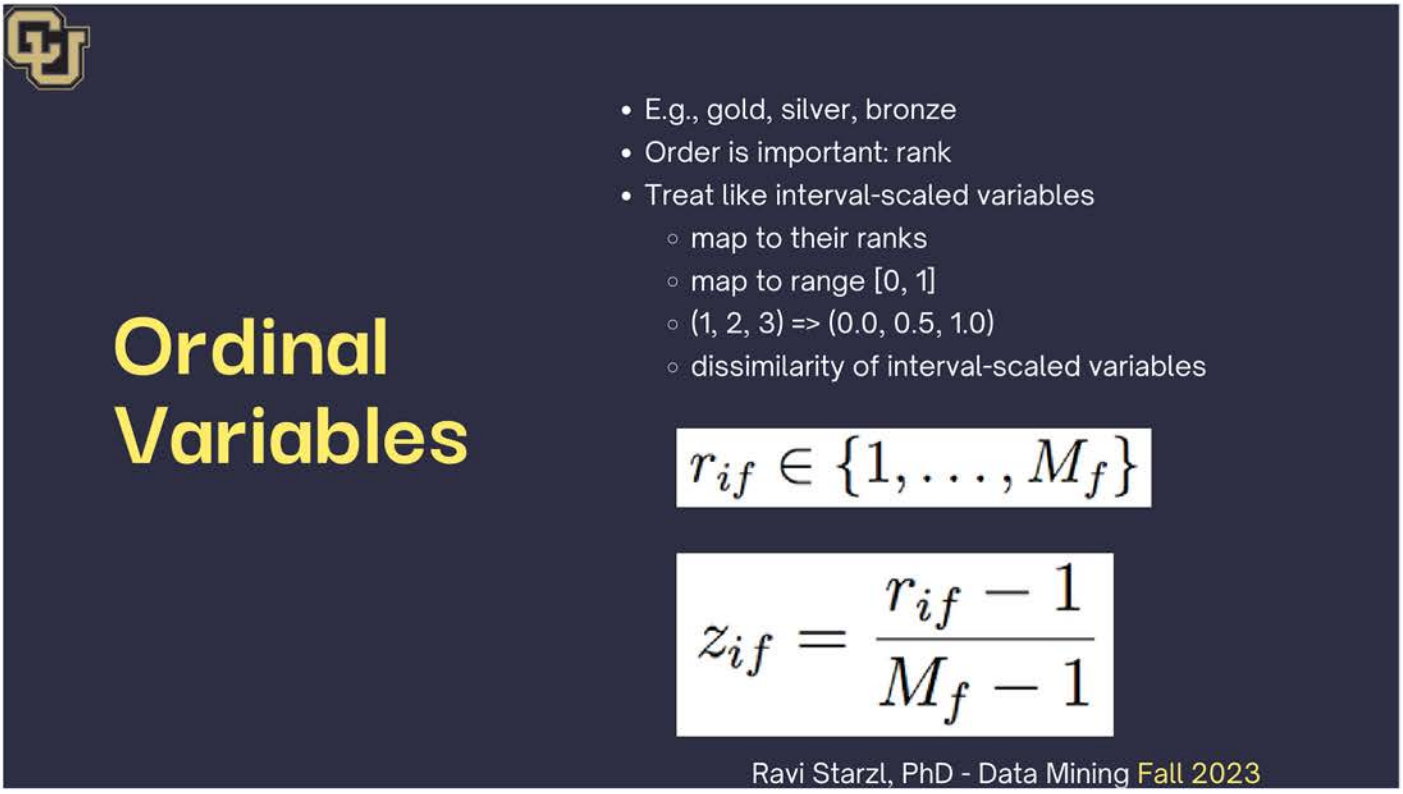
Applying this formula to our data, the dissimilarity between Jack and Mary can be calculated as $d(Jack, Mary) = (0+1)/(2+0+1) = 0.33$. This calculation shows that Jack and Mary have a difference in one attribute and agree on two attributes, resulting in a fairly low dissimilarity value of 0.33.

When we analyze Jack and Jim, the dissimilarity is calculated as $d(Jack, Jim) = (1+1)/(1+1+1)$ =

0.67. This outcome reveals a greater disparity between Jack and Jim, as they differ on two attributes but agree on only one.

Furthermore, calculating the dissimilarity between Jim and Mary gives us d(Jim, Mary) = (1+2)/(1+1+2) = 0.75, indicating the highest level of dissimilarity amongst the three comparisons.

Utilizing such analysis can be advantageous in various domains. For instance, in the realm of recommendation systems, recognizing user dissimilarities can be instrumental in offering personalized suggestions. Likewise, in social network analysis, it aids in discerning relationships between individuals. Evaluating the dissimilarity in responses to specific queries allows us to glean valuable insights into the group under study.



Ordinal variables refer to a specific kind of categorical variable where the distinct categories maintain a natural sequence or hierarchy. A straightforward instance to illustrate ordinal variables can be found in competition medals – gold, silver, and bronze. In this scenario, the medals represent distinct categories that possess an intrinsic ranking of value, with gold being superior to silver, and silver surpassing bronze in terms of ranking.

A primary attribute of ordinal variables is the importance of the order, contrasted with the indifference toward the exact difference between the values. Referring back to the medal example, it is clear that a gold medal signifies a higher achievement than a silver, and a silver outshines a bronze. However, it is not feasible to accurately measure the precise degree of superiority, indicating that the value difference between gold and silver might not equate to that between silver and bronze.

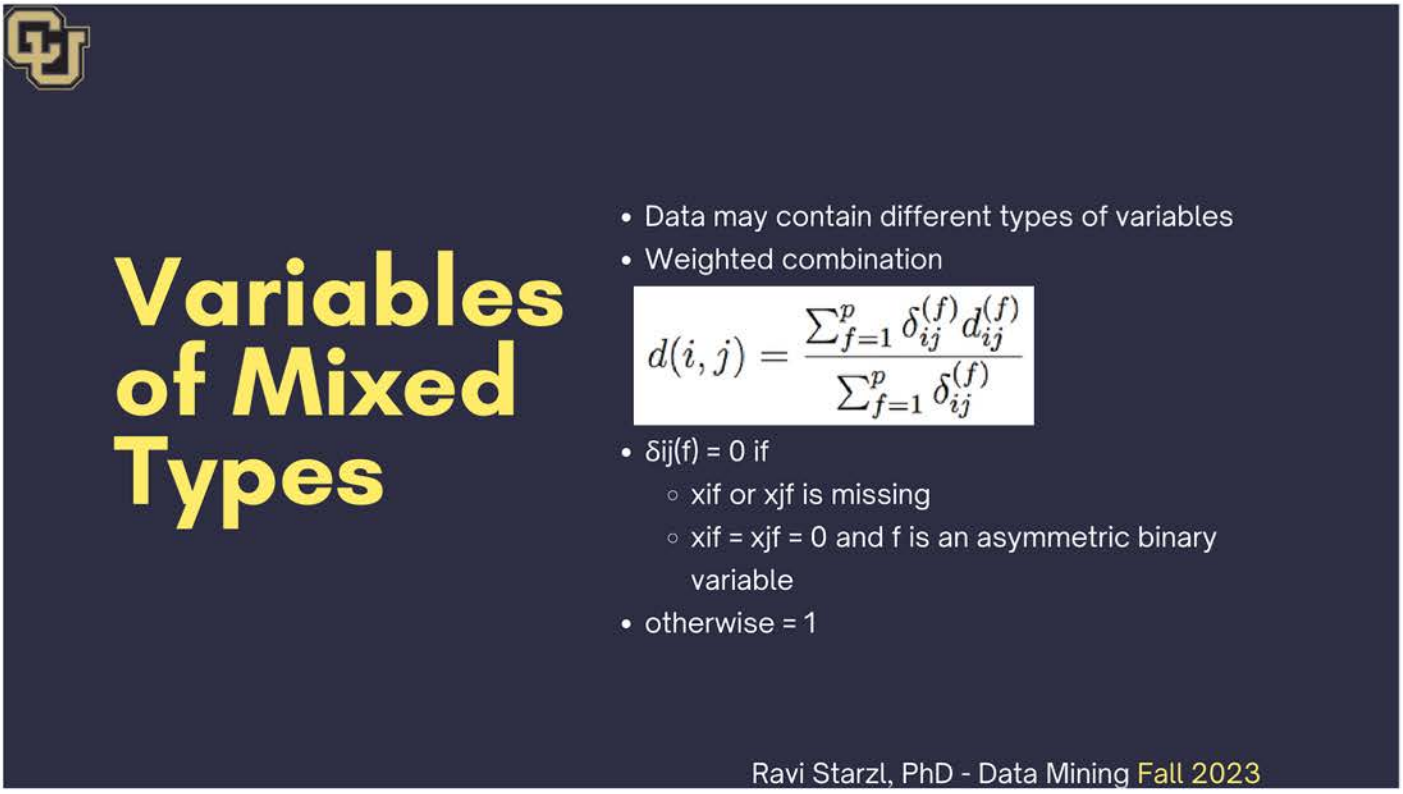
In the realm of data analysis, ordinal variables are often handled as if they were interval-scaled variables. This transformation can be conducted in several ways. A rudimentary method involves assigning ranks to these variables. In the context of the medals, gold would be assigned a rank of 1, followed by silver with a rank of 2, and bronze with a rank of 3.

A further strategy entails the normalization of these ranks so that they adhere to a specific range, commonly [0, 1]. This modification is beneficial, as it converts the variable into a format that is conducive to a range of analyses. Applying this to the medal example, the modified ranks would be assigned as follows: gold would correspond to 0.0 (minimum rank), silver to 0.5, and bronze to 1.0 (maximum rank). This normalization procedure aids in mitigating the potential influence of

scale discrepancies between variables on the analysis outcomes.

To compute dissimilarity utilizing these normalized values, they are perceived as interval-scaled variables, and formulas typically reserved for these variables are employed. For instance, in a case involving two athletes, A and B, with gold and silver medals respectively, the dissimilarity in terms of medal ranking could be assessed as the absolute disparity between their normalized ranks, equating to |0.0 - 0.5| = 0.5 in this case.

Ordinal variables are vital given their prevalence across diverse domains. Ranging from gathering feedback in surveys (like assigning a product rating from 1 to 5) to documenting competition outcomes (as demonstrated in the medal example), they furnish invaluable perspectives when subjected to appropriate data analysis.



In the analysis of real-world data, analysts often encounter a variety of variable types, including continuous, binary, ordinal, and nominal. One of the challenges that arise in this context is the measurement of similarity or dissimilarity between data points, given the diversity in variable types. Each variable type necessitates its own specific method for gauging similarity or dissimilarity, leading to the pivotal question: how can these diverse measures be amalgamated into a single coherent measure?

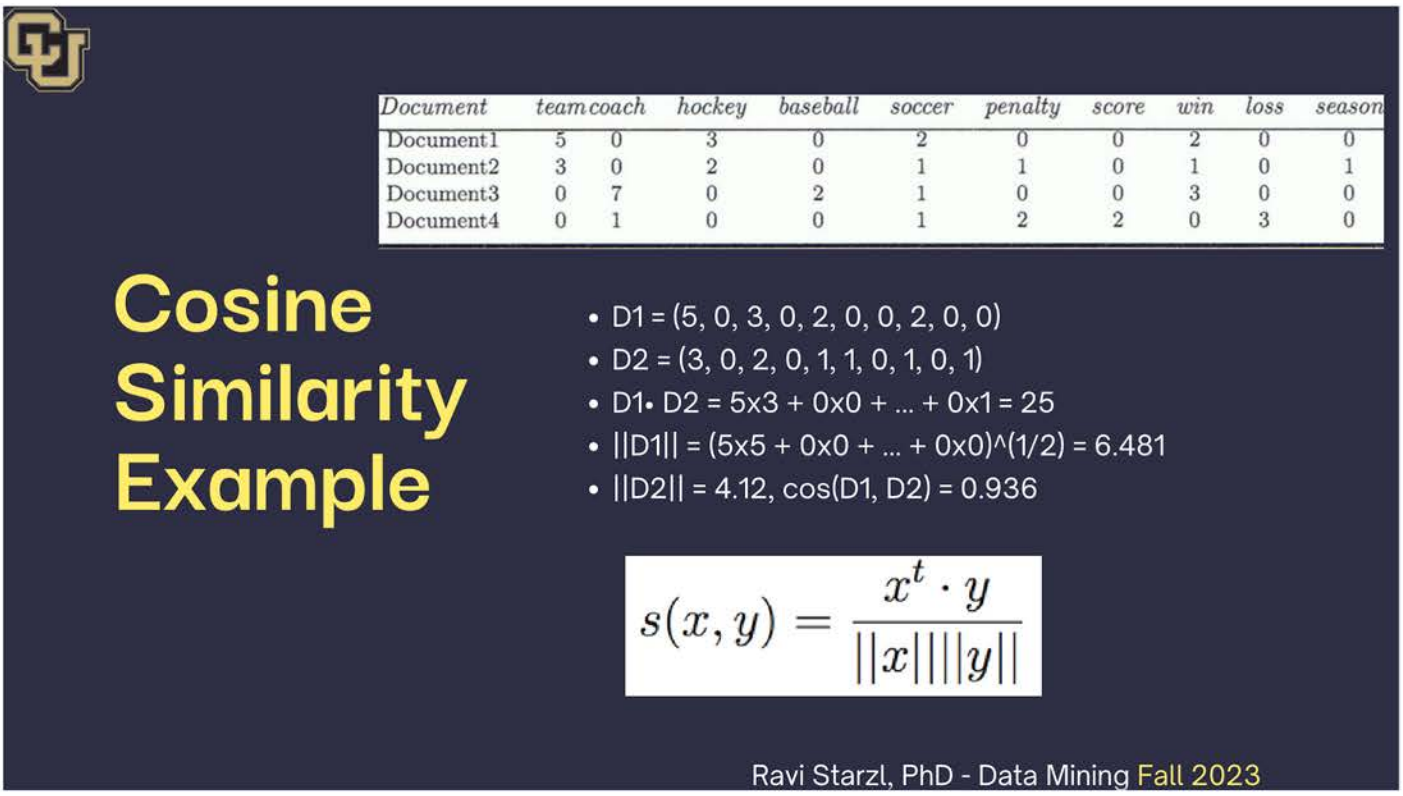
The notion of weighted combination addresses this as a viable solution. This approach operates on the simple premise that variables do not hold equal significance in determining similarity or dissimilarity. By allocating weights to each variable, based on its estimated significance, a unified measure can be formulated that aptly recognizes the varying contributions of each variable.

To illustrate, envision a scenario where the task is to ascertain the similarity between two students based on two variables: the grade achieved in a math course (a continuous variable) and their major (a binary variable - either in computer science or not). Suppose the grade is regarded as having double the importance of the major in establishing similarity, thus warranting a weight of 2 for the grade and 1 for the major.

The process extends beyond merely assigning weights. It is imperative to also discern the appropriate manner to calculate the individual similarity or dissimilarity for each type of variable. For continuous variables such as grades, a straightforward difference might suffice, while binary variables, like the major, may necessitate a function that yields 0 when the majors are identical and 1 when they differ.

The proposed formula, denoted as δij(f), facilitates the computation of this weighted combination. In this formula, δij(f) indicates the dissimilarity between entities i and j concerning variable f, and xif and xjf denote the respective values of variable f for entities i and j. If any data point is missing, or both values equate to zero for an asymmetric binary variable (a situation where a non-zero value holds more significance than a zero value, analogous to our major example), the dissimilarity registers as 0; otherwise, it is 1.

The fundamental objective of this strategy is to distill a complex, multidimensional comparison into a single, comprehensible metric. Despite the potential complexity in computation, the core concept remains straightforward - it aims to integrate diverse data elements into a cohesive unit.



Cosine similarity is a mathematical technique utilized to calculate the similarity between two non-zero vectors, frequently applied in fields like text analysis, recommendation systems, and various data mining undertakings. This method calculates the cosine of the angle between the two vectors in question. A notable advantage of this approach is that the resulting similarity measurement is independent of the magnitude of the vectors, focusing solely on their orientation.

Consider the example illustrated with two vectors, D1 and D2, which hypothetically represent the varying interests of two individuals across a range of topics. In this representation, a value of 0 signifies no interest, while progressively higher values indicate an increasing level of interest. Here are the vectors:
D1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
D2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

To interpret, the individual components of these vectors might symbolize interest in specific subjects, like mathematics, history, and music, designated by the first, third, and fifth components, respectively. From the data, it's apparent that D1 expresses a heightened interest in the topics represented by the first, third, fifth, and eighth components, whereas D2 displays some engagement with the topics denoted by the sixth and tenth components, areas where D1 shows no interest.

To compute the cosine similarity, we initiate by calculating the dot product of the vectors. This is achieved by summing the products of their respective components, resulting in a value of 25. Subsequently, we determine the magnitude or norm of each vector, which is the square root of the

sum of the squares of each component. The calculations yield values of 6.481 for D1 and 4.12 for D2.

The final step involves dividing the dot product by the multiplication of the magnitudes of the two vectors, which gives us the cosine similarity between D1 and D2:
cos(D1, D2) = 25 / (6.481 * 4.12) = 0.936

This outcome, 0.936, represents the cosine similarity between the two vectors, wherein the possible range of values extends from -1 to 1. A score of 1 indicates identical vectors, -1 points to completely dissimilar vectors, and a score of 0 suggests that the vectors are orthogonal or unrelated. In this scenario, a cosine similarity value of 0.936 implies a substantial level of similarity between the interests denoted by vectors D1 and D2.

In this analytical exercise/reflection, examine the distinctive attribute types that one might find in various datasets, including those from platforms like Twitter, or fields such as sports, news, and traffic management. These datasets are characterized by their unique contexts, each containing a specific structure and kind of data.

In the context of Twitter, a single tweet encompasses several attributes: the text content, user handle, timestamp, geolocation, and hashtags. These attributes denote different kinds of data, where the text content is nominal, the timestamp is interval-scaled, geolocation might be ratio-scaled, and the hashtags are binary, indicating their presence or absence in a tweet.

A sports dataset includes a range of attributes like player names, teams, scores, timestamps, match locations, and player statistics. In this dataset, both the player names and teams fall under the category of nominal attributes. The scores are ratio-scaled, timestamps are interval-scaled, while the locations are nominal. Player statistics might incorporate both ratio and interval-scaled at tributes.

When analyzing a news dataset, one might find attributes such as the article title, author, publication date, text of the article, and categories. Here, both the titles and authors are nominal, the publication date is interval-scaled, the text is nominal, and the categories function as binary att ributes.

A traffic dataset might be comprised of attributes such as the location, timestamp, count of vehicles, type of vehicles, and their average speed. In this case, the location is a nominal

attribute, the timestamp is interval-scaled, the count of vehicles is ratio-scaled, vehicle type is nominal, and the average speed is also ratio-scaled.

The potential knowledge garnered from these datasets can vary significantly. For instance, analysis of Twitter data through sentiment analysis might unveil trends in public opinions on specific topics. Sports datasets can provide an in-depth look into the performance of teams and efficiency of players. Scrutinizing news datasets could help in pinpointing recurring themes and potential biases, whereas traffic datasets can be instrumental in identifying peak traffic times, common congestion areas, and evaluating the effectiveness of traffic management strategies.

The implications of this acquired knowledge span multiple domains. Insights garnered from Twitter can facilitate informed decisions in policymaking, marketing, and investment. Sports data analysis can contribute to strategic team selection, fostering player development, and enhancing game strategies. Analysis of news data can potentially enhance news recommendation systems, assist in journalistic research, and shape public relations strategies. Meanwhile, traffic data can be pivotal in urban planning, optimizing traffic control, and making informed infrastructure development decisions.

A fundamental step in effective data analysis is the identification of attribute types present within our dataset. A comprehensive understanding of these attributes aids in the selection of suitable measures of similarity, streamlining preprocessing steps, and opting for the most effective data mining techniques.

Preprocessing is a fundamental initial step in data mining that encompasses the modification of raw data into a format that is both comprehensible and apt for further analysis. This procedure not only facilitates the later application of algorithms more efficiently but also enhances their accuracy. The primary focus here is an elaboration on the components of data preprocessing, namely an overview of data preprocessing, assessing data quality, and delineating the principal tasks involved in data preprocessing such as data reduction, data cleaning, data integration, data transformation, and discretization.

A pivotal aspect to consider during data preprocessing is the quality of data. This term signifies the condition of qualitative or quantitative data elements. Data is considered of high quality when it is aligned with its intended applications in operations, planning, and decision-making, showcasing attributes such as reliability, precision, completeness, and trustworthiness. It is essential to assess data based on multiple criteria including accuracy, which gauges how well the data represents the real-world entities or events it is describing; completeness, which assesses the inclusion of all necessary data; consistency, which ensures uniformity throughout the dataset; timeliness, which evaluates the currency of the data; believability, which measures the degree of trust users have in the data; interpretability, which checks if data is presented in understandable formats and units; and accessibility, which appraises the ease with which the data can be acquired and utilized.

A detailed exploration of data preprocessing reveals several major tasks, including data cleaning, data integration, data reduction, and data transformation. Data cleaning is concerned with addressing issues of missing and noisy data, and rectifying inconsistencies, which can involve

strategies such as substituting missing values with the attribute mean or removing outliers. Data integration refers to the amalgamation of data from disparate sources into a coherent data repository, necessitating the management of redundant and inconsistent data, and schema integration. Data reduction aims to diminish the data volume while retaining the integrity of the analytical results, utilizing techniques such as dimensionality reduction and numerosity reduction. Data transformation and discretization, on the other hand, ready the data for mining processes through operations like normalization, scaling, and aggregation, and involve the conversion of continuous data into categorical forms.

Preprocessing serves as the precursor to data mining, significantly shaping the outcomes of the mining process.

In data mining, assessing the quality of data is a fundamental task, serving to determine the suitability of the data for ensuing analysis. This involves the evaluation of several critical data quality measures: accuracy, completeness, consistency, timeliness, believability, interpretability, and accessibility.

Accuracy is a measure that signifies how close a data value is to its actual or true value. A practical illustration of this could be recording students' heights in a class setting; the accuracy of the data is contingent upon the recorded heights mirroring the actual heights of the students. The presence of faulty data entries or incorrect measurements might induce inaccuracies.

Completeness, as implied by its name, is concerned with ensuring that all necessary data is accounted for. If the height details of certain students are absent in the data, it denotes incompleteness. Such gaps in data can precipitate biased interpretations and skewed results, making it imperative to pursue comprehensive data collection.

Consistency is centered around maintaining uniformity within the data, implying that the data should be contradiction-free and uniformly presented across all instances. For instance, maintaining a consistent date format across all entries in a dataset is a manifestation of this quality measure.

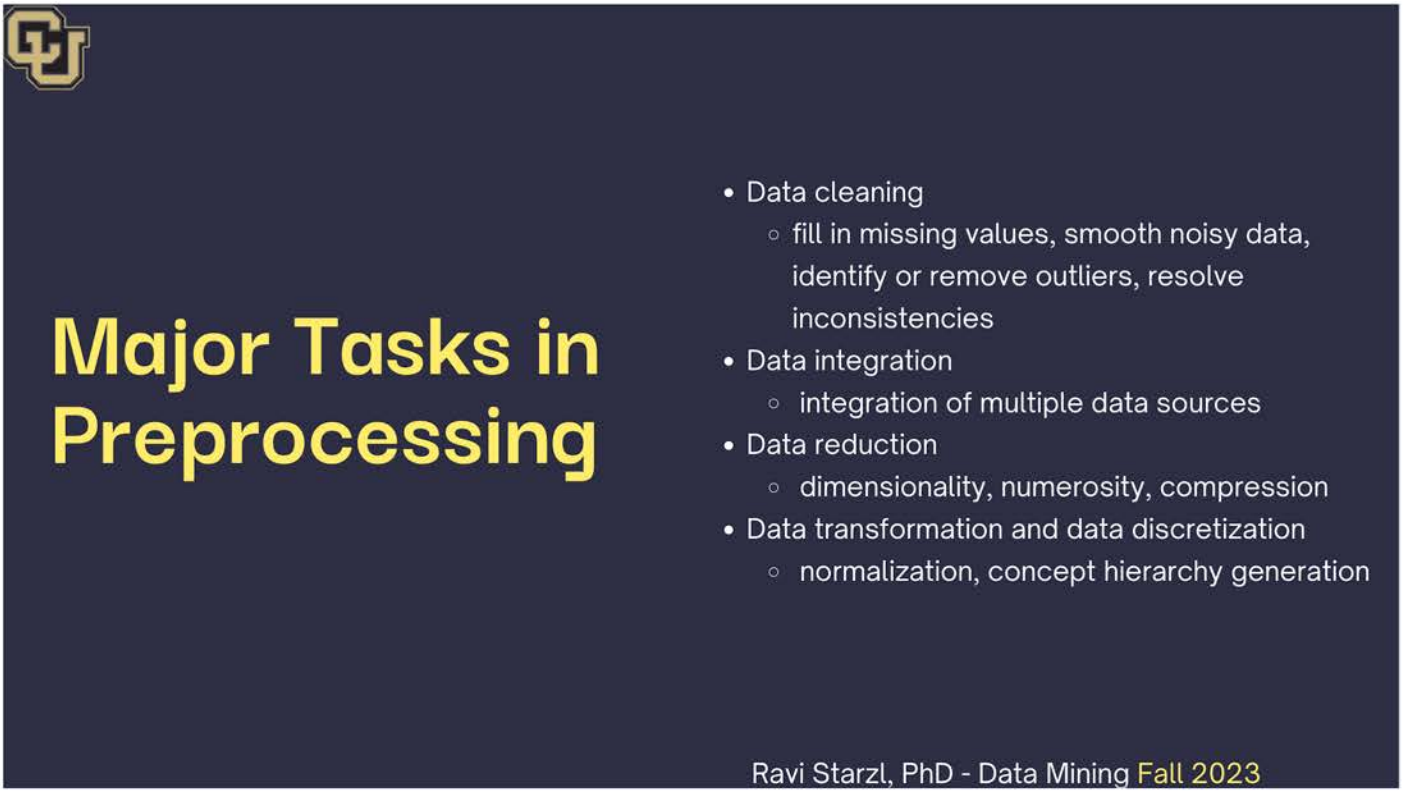
Timeliness denotes the readiness of data availability as required. Utilizing outdated data might not represent the current status accurately, potentially leading to inaccurate insights or decisions.

Believability evaluates the degree to which data is perceived as credible and truthful. The credibility often hinges on the source and the data collection methodology. For instance, a recorded height of 10 feet for a student would not be deemed believable, considering the standard height parameters for humans.

Interpretability involves the ease of data understanding and comprehension, extending beyond mere readability to ensure logical interpretation and coherence. For instance, if a dataset employs codes to denote various categories, it should also furnish a key to elucidate the meaning of each c ode.

Accessibility, the final measure, refers to the simplicity involved in retrieving and utilizing the data. Despite being accurate, complete, and reliable, data's utility can be significantly hampered if it is not readily accessible or is formatted in a way that complicates usage.

Adhering to these measures is important in evaluating data quality in data mining. Failure in fulfilling these criteria can result in suboptimal mining outcomes, thereby potentially leading to incorrect decisions and interpretations.



Here we elucidate some techniques utilized during data preprocessing in the context of data mining. These strategies are essential in enhancing data quality and priming it for the subsequent mining phase.

Data cleaning entails the identification and rectification or elimination of errors or discrepancies present in the data. For example, in a car sales dataset, if certain entries in the 'Price' column are populated with non-numeric values such as 'N/A' or 'Unknown', it necessitates data cleaning to either substitute these entries with meaningful values or expunge the rows embodying these value s.

Following this, data integration is performed to amalgamate data from diverse sources into a cohesive data repository. This can be illustrated with a multinational enterprise that maintains customer information dispersed across several regional databases. The amalgamation of this data into a consolidated data warehouse facilitates a more profound and insightful analysis.

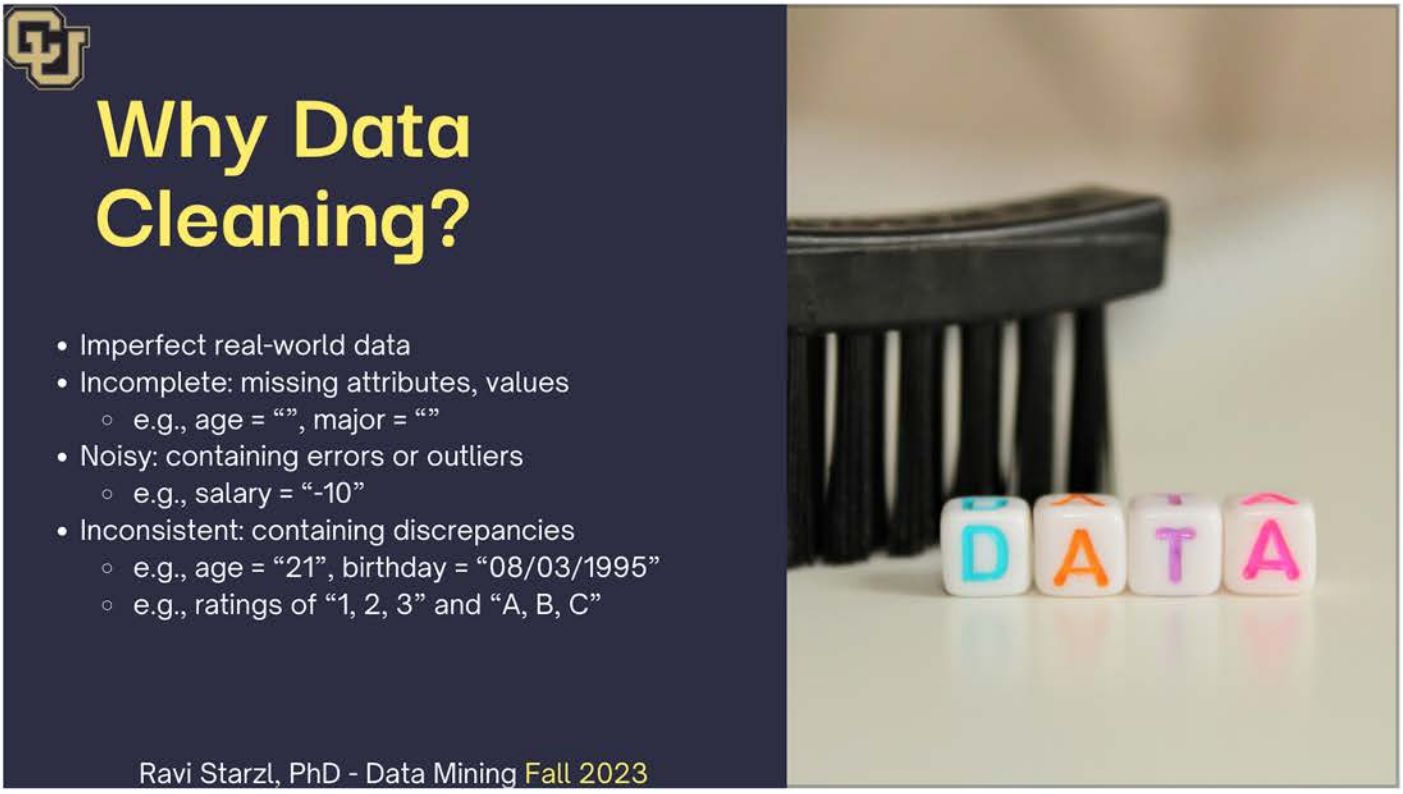
Data transformation, the subsequent step, involves the modification of raw data into a prescribed format that is more amenable to analysis. A case in point would be the normalization of data, whereby it is adjusted to a specified range, enhancing the performance of several machine learning algorithms. For instance, normalizing the 'Income' column values, which span from $20,000 to $2,000,000, to lie between 0 and 1 can be a prudent approach.

Data reduction is employed to condense the data whilst preserving its essence and avoiding the introduction of substantial inaccuracies. An example of this is dimensionality reduction, where

techniques such as Principal Component Analysis (PCA) are utilized to decrease the data's dimensionality by converting the original attributes into a lesser set of features, thereby maintaining the majority of the information.

Data discretization is applied to change continuous variables into their categorical equivalents, which can be advantageous in certain analyses. For instance, categorizing customers into distinct age groups like '18-24', '25-34', etc., can sometimes be more beneficial compared to using their exact ages during a marketing survey.

Each of these techniques holds a distinct role in data preprocessing, their application dictated by the unique demands of the data mining activity being undertaken. These strategies are instrumental in augmenting the data quality, thereby enhancing the results of data mining.



Data, acquired from a variety of sources, often contains flaws that hinder immediate analysis. Here, we elucidate why data cleaning is a critical phase in the data analysis process.

Initially, one encounters the problem of incompleteness in real-world data. This data frequently has missing attributes or values, obstructing smooth analysis. A common instance would be finding datasets with absent information like the age or primary field of study of some individuals, making precise interpretation of the data elusive. These missing values might arise due to malfunctioning data collection tools, unavailable data at the time of collection, or certain data points being irrelevant to specific individuals.

Data can sometimes be noisy, characterized by errors or outliers that potentially alter the comprehensive understanding of the data. An illustrative example is a data entry indicating an individual's salary as "-10", a blatant error since salary cannot hold a negative value. Such discrepancies might originate from human errors, system glitches, or improper data input. The task of pinpointing these inconsistencies, albeit difficult, is fundamental to prevent analysis results from being skewed.

Adding to the complexities is the issue of inconsistency, where the data contains conflicting information, complicating the analysis process. For instance, an inconsistency arises when an individual's age is noted as "21", yet their birth date is recorded as "08/03/1995", thereby implying an age older than 21 in most years. Moreover, discrepancies in rating systems, like differing scales, can foster confusion and incorrect comparisons.

To mitigate these concerns, data cleaning serves as a potent tool. It facilitates the rectification of missing data using methods like imputation, enabling the supplementation of gaps with plausible estimates. Noisy data can be refined using statistical techniques, minimizing the repercussions of errors or outliers. Likewise, inconsistent data can be harmonized using data transformation strategies.

In order to fully grasp the concept of data imperfections, it is necessary to get into the underlying reasons that contribute to data being imperfect.

Incomplete data is a common issue, often evidenced by "not applicable" values in databases or datasets where certain attributes or characteristics do not pertain to all entries. For instance, in a dataset of students, the field designated for 'company name' might remain empty or be labeled as 'not applicable' for students who are not currently employed.

The gap in time between the collection and analysis of data significantly contributes to data incompleteness. Given the dynamic nature of real-world data, substantial delays between data collection and analysis can result in discrepancies. Taking the example of a company revenue dataset, if analyzed after a prolonged period since its collection, the data might not encapsulate recent fluctuations in the company's revenue, rendering the data incomplete.

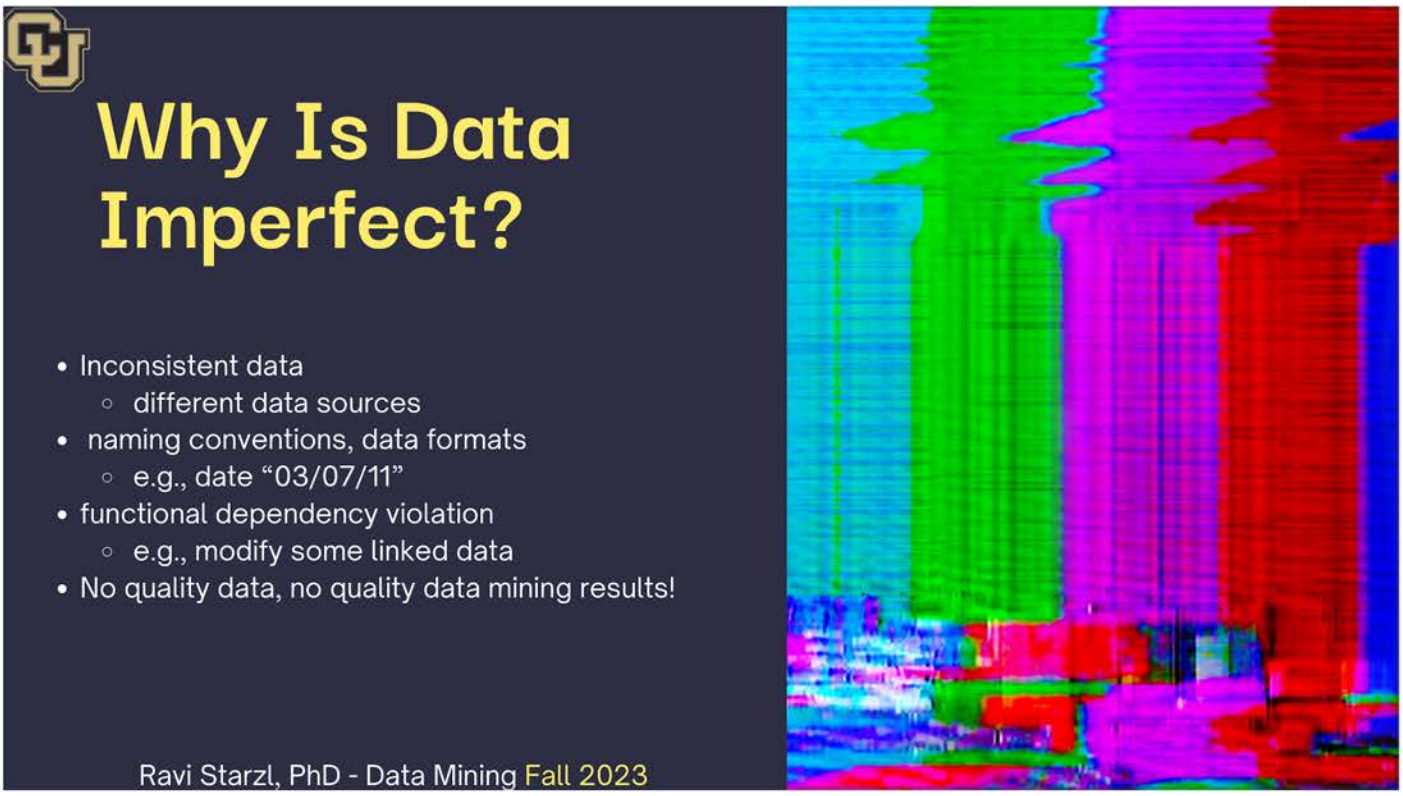
Complications stemming from human error or issues with hardware or software can also lead to data gaps. Imperfections may arise from bugs in data scraping software, network disruptions during data transfer, or human errors ranging from simple oversights in form completion to more complex misunderstandings of data entry protocols.

In noisy data, we encounter issues such as inaccurate data collection instruments which can generate erroneous or 'noisy' data. In a temperature study, for instance, uncalibrated thermometers can record incorrect data. Further, data noise can be introduced through errors occurring during data entry, either by humans or computers. These might involve a cashier

recording an incorrect transaction amount or glitches in data entry software.

Data transmission errors are prevalent when data is disseminated electronically. For example, data corruption can occur if an internet connection is lost while transmitting a substantial dataset, resulting in 'noisy' or compromised data at the recipient's end.

Data quality can be compromised by various factors, making it critical to acknowledge these potential issues and implement suitable data cleaning strategies. This proactive approach ensures the reliability of data analysis outcomes, facilitating more accurate and informed decision-making.



While we have already examined the issues of incomplete and noisy data, it is equally vital to address the challenge posed by inconsistent data, which can stem from various factors.

A primary cause of inconsistency is the diversity of data sources utilized in the data collection process. The data could be amassed from a range of platforms including databases, APIs, surveys, among others. Each of these platforms typically operates based on its own set of standards, formats, and structures. For instance, while analyzing patient data from different hospitals, one might encounter disparate systems for recording age – one hospital might document age in years and months, whereas another might only note down the birth year. The merger of these datasets without proper reconciliation would inevitably lead to inconsistent data.

Parallel to the issue of varied data sources is the divergence in naming conventions and data formats. Different systems might adhere to different protocols for recording customer names; one might follow a 'first name, last name' convention, whereas another opts for a 'last name, first name' approach. Moreover, the representation of dates can also vary considerably between databases, potentially leading to confusion and inconsistencies when amalgamating data from different sources. For instance, the date "03/07/11" could either denote March 7, 2011, or July 3, 2011, contingent upon the utilized format.

Inconsistencies in data can emerge due to the breach of functional dependencies, which are established constraints between two sets of attributes in a database. This discrepancy arises when a modification in linked data, such as altering the 'EmployeeName' associated with a specific 'EmployeeID', is not uniformly updated across all instances, thereby violating functional

dependencies and engendering inconsistency.

These inconsistencies, akin to missing or noisy data, significantly diminish data quality. There exists a consensus in the field of data science encapsulated by the phrase 'Garbage In, Garbage Out', highlighting that the output quality is intrinsically linked to the input data quality. Consequently, inconsistencies in data can render even the most sophisticated data mining algorithms ineffective, yielding unreliable results. This accentuates the necessity for thorough data cleaning prior to initiating any data analysis or mining endeavors.

Recognizing the origins of data imperfections and adeptly mitigating them through comprehensive data cleaning processes is critical in securing dependable and high-quality outcomes in data mining.

In the process of data cleaning, addressing the issue of missing data is a common yet critical task.

An initial, straightforward strategy could be to simply disregard the tuple, which entails eliminating any entry in the dataset that exhibits one or more missing values. While this method might seem convenient, it is not always advantageous. Particularly in cases where the dataset is relatively small or the incidence of missing values is prevalent, employing this strategy might result in a considerable depletion of vital information. Nevertheless, this method retains its relevance when the quantum of missing data is minor and exhibits a random distribution pattern.

Alternatively, one might opt to manually input the missing values. This strategy might be feasible for datasets of a smaller magnitude but tends to be both unfeasible and susceptible to errors when applied to larger datasets. Nonetheless, in specific instances, leveraging domain knowledge or conducting supplementary research can facilitate the effective completion of these data gaps.

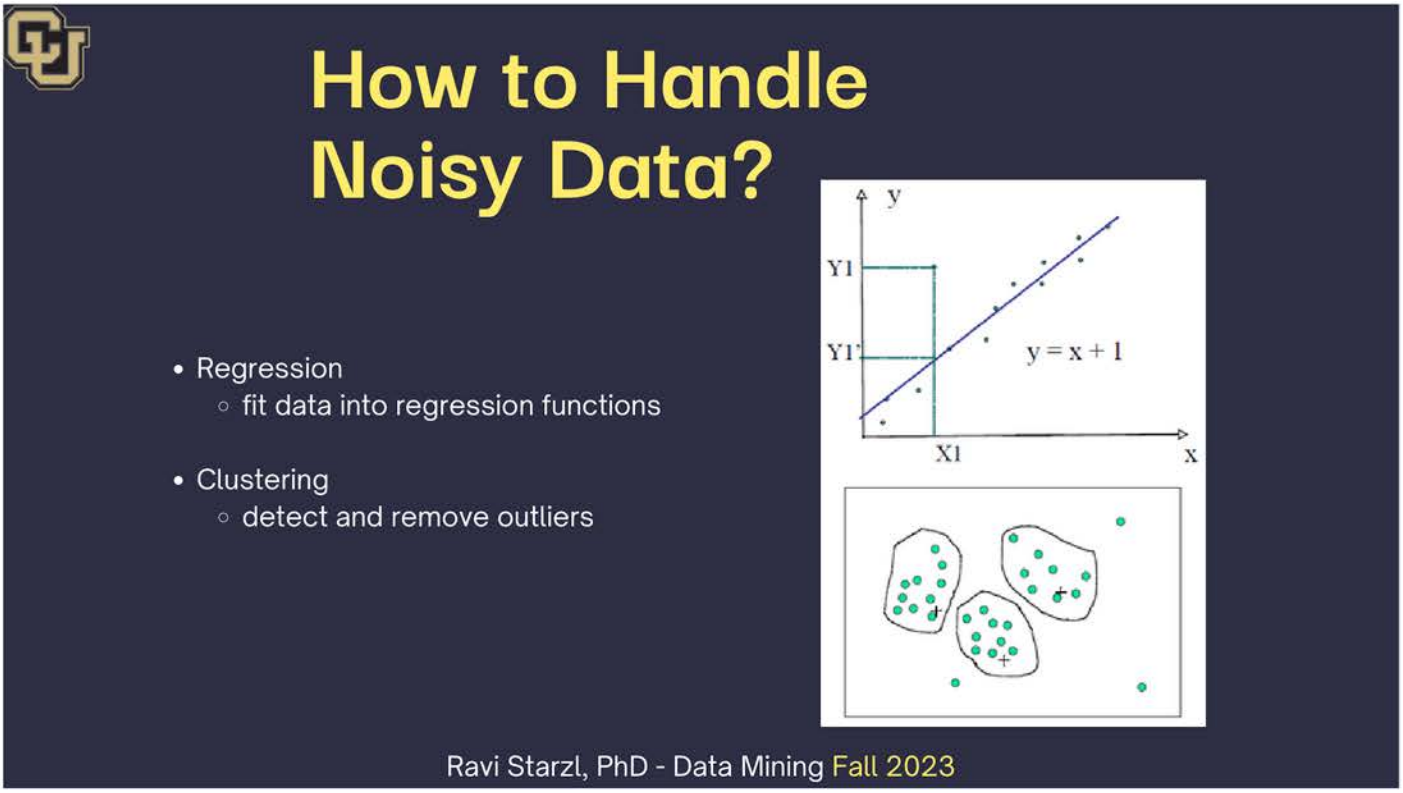
Transitioning to more structured approaches, an automated system for filling missing data can be implemented. This could involve utilizing a global constant, a unique value that is not represented within the attribute domain, to denote the absence of data, rather than approximating the missing values. Another automated approach focuses on employing the attribute mean for filling the gaps, where the missing value is replaced with the average of the existing values pertaining to that attrib ute.

Leveraging the attribute mean specific to the same class is another viable strategy, especially

when the dataset exhibits clear categorical distinctions. An application of this method can be seen in scenarios like estimating the missing weight of an animal in a zoo dataset, where the average weight of other entities within the same category is used to fill the gap.

Utilizing sophisticated techniques to ascertain the most probable value to replace the missing data can be a resourceful method. Implementing techniques such as regression analysis, Bayesian inference, or decision trees enables the estimation of the most likely value for a missing attribute, using various other correlated attributes as a reference.

Selection of an appropriate strategy for addressing missing data is significantly influenced by multiple factors including the context of the study, the characteristics of the data, the prevalence of missing values, and the ultimate objective of the analysis. A thoughtful consideration of these aspects, perhaps even integrating multiple methods, could pave the way for a more proficient handling of missing data.



Noisy data, characterized by errors or outliers, can undermine the accuracy and reliability of data analysis efforts. Let's explore two prevalent methods to address noisy data: regression and clusteri ng.

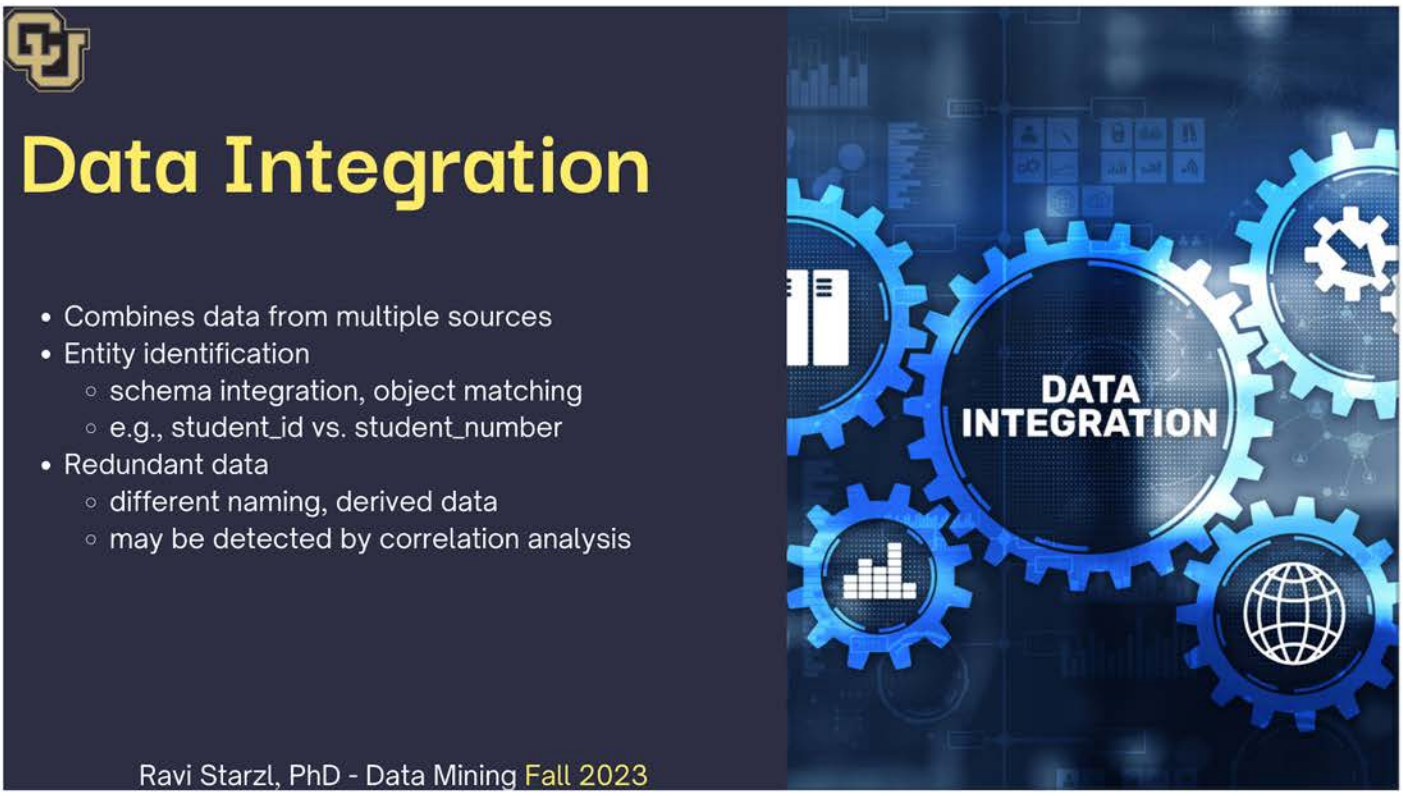
Starting with regression, this method endeavors to align data with regression functions, facilitating the prediction of one attribute based on others. A prime example of this is simple linear regression, which seeks to determine a straight line that most accurately represents the distribution of data points. To exemplify, consider analyzing a dataset consisting of individuals' weights and heights. Within this dataset, there might be data points that diverge substantially from the majority, potentially representing errors or outliers. In such instances, regression can be utilized to craft a line that forecasts weight using height as a variable, thereby allowing the identification of outliers as points notably distant from this line. Recognizing these outliers is a pivotal step, either pointing towards erroneous entries to be rectified or removed, or highlighting unique cases warranting deeper investigation.

Transitioning to clustering, it serves as an unsupervised machine learning algorithm that aggregates data points based on similarity metrics, proving particularly adept at identifying and removing outliers. For instance, consider a dataset encapsulating shopping habits, encompassing attributes like the number of purchased items, overall expenditure, and shopping frequency. Implementing a clustering algorithm such as k-means can reveal distinct clusters epitomizing different spending brackets, namely low, medium, and high spenders. However, certain data points may conspicuously deviate from these clusters, signaling potential outliers. Anomalies such as extremely high item purchases coupled with minimal total expenditure might flag data

recording or processing errors. Utilizing clustering to pinpoint these outliers serves as a beneficial strategy to enhance data quality.

In practice, the employment of regression and clustering is often visually represented through scatter plots. A plot illustrating regression portrays data points complemented by a line delineating the optimal fit, while a clustering plot presents data points grouped into clusters, with outliers discernibly isolated from these congregations.

It is essential to manage noisy data during the data preprocessing phase. Regardless of the approach chosen - be it regression, clustering, or alternate techniques - the overarching objective remains to isolate and rectify inaccuracies, thereby safeguarding data integrity and bolstering the reliability of ensuing analysis and model development.



Data integration constitutes an important process in the realm of data science, especially in the current big data era where data is sourced from a myriad of avenues such as databases, web scraping, and IoT devices, among others. The primary objective is to establish a consolidated viewpoint of these varied data sources, a task that presents considerable challenges due to obstacles such as entity identification and data redundancy.

Firstly, entity identification emerges as a fundamental hurdle in the data integration process. Consider the scenario of assimilating data from two student databases belonging to different departments within a university. While one database employs the attribute label 'student_id', the other utilizes 'student_number', both denoting the same entity, the unique identifier of a student, albeit under distinct nomenclatures. This scenario exemplifies schema integration, necessitating the unification of schemas from disparate data sources. Consequently, the alignment of attributes like 'student_id' and 'student_number' embodies object matching, which facilitates the synchronization of analogous objects across various datasets.

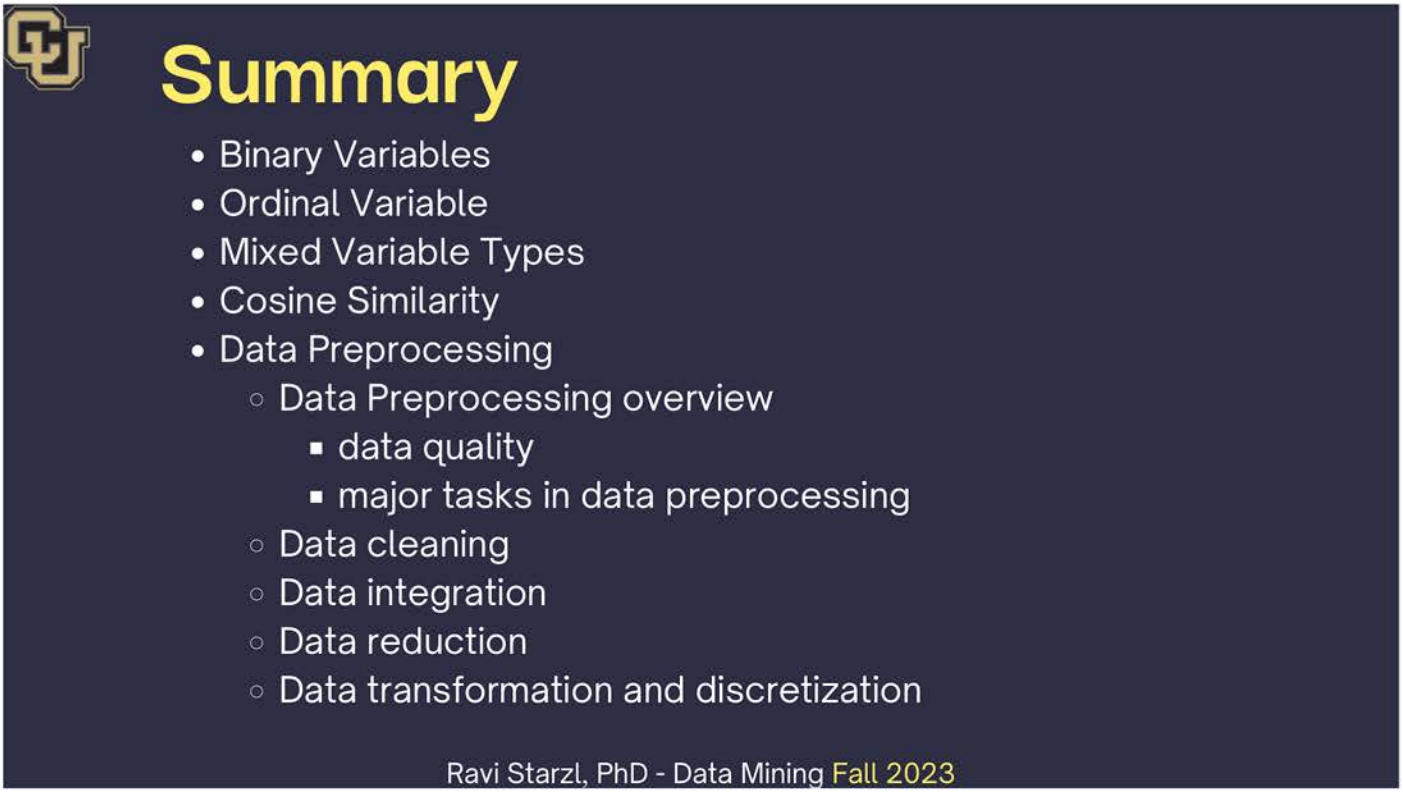
Taking another scenario into account, let us examine the integration of data derived from the sales and customer support wings of an e-commerce enterprise. Here, the sales database employs 'customer_id' as the primary key, while the customer support database opts for 'customer_reference_number'. Recognizing that both these attributes denote the same entity - the customer - is vital to ensuring seamless data integration across the two datasets.

Transitioning to the second predominant issue, data redundancy tends to be a recurring complication when integrating data from diverse sources. Redundancy manifests either when

identical data is depicted in varied manners or when an attribute is inherently dependent on another. For instance, a dataset may illustrate 'total_price' in both US dollars and Euros, which, albeit beneficial in certain scenarios, generally signifies data redundancy, fostering potential inconsistencies if not simultaneously updated. Additionally, redundancy is evident in cases of derived data, such as a retail transaction dataset incorporating attributes 'item_price' and 'quantity', along with a 'total_price' attribute, calculable as their product, thereby denoting redundant data that can be computed from existing attributes as necessary.

An efficient strategy to pinpoint redundant data is correlation analysis. A high correlation between two attributes generally signifies overlapping information, hinting at potential redundancy. For instance, a dataset detailing housing attributes might reveal a strong correlation between 'house_size_in_sq_ft' and 'number_of_rooms', given the tendency for larger houses to encompass more rooms. Recognizing this, analysts might opt to retain only one of these attributes in their analysis to circumvent data redundancy.

Data integration is a complex procedure demanding meticulous management of aspects such as entity identification and data redundancy. Despite the inherent challenges, adept data integration fosters a comprehensive and accurate representation of the available data, paving the way for optimized data mining outcomes.



In this lecture, we have explored various critical aspects of data science and preprocessing. Let us summarize and review some of these important concepts.

Initially, we focused on binary variables, which are characterized by only two possible outcomes: such as 'yes' or 'no', 'true' or 'false', and 'success' or 'failure'. These variables are immensely beneficial in data science due to their simplicity in analysis and interpretation. They serve as the foundation of numerous predictive models, particularly in classification problems where the objective is to predict a binary outcome, such as determining if an email falls under the category of spam.

Subsequently, we examined ordinal variables that depict order or hierarchy within a specified set. Examples of these are categories like 'low', 'medium', and 'high' or rating scales ranging from 1 to 5. The main obstacle with these variables is that although the order of categories is clear, the exact differences between them remain undefined. Certain techniques, such as ordinal logistic regression, are tailored to address these kinds of variables.

Analyzing real-world data often brings the complexity of managing mixed variable types to the forefront. It is not uncommon for a single dataset to incorporate binary, ordinal, categorical, and continuous variables. Addressing this variability necessitates preprocessing and occasionally diverse analytical approaches for each variable type.

We also covered the notion of cosine similarity, a metric used to determine the similarity between two vectors. This concept finds extensive applications in text analysis, specifically in evaluating

document similarity. The cosine similarity is computed as the cosine of the angle between two vectors, providing a measure of similarity irrespective of their magnitude.

Data preprocessing encompasses a range of tasks geared towards enhancing data quality before undertaking analysis. This step includes tackling issues related to data quality, including incomplete, noisy, and inconsistent data elements.

Data preprocessing can be broken down into several pivotal tasks: data cleaning, data integration, data reduction, and data transformation and discretization. Data cleaning is the process of addressing data 'dirtiness' by managing missing values, reducing noise, and rectifying inconsistencies. Data integration, on the other hand, involves merging data from varied sources into a coherent dataset, as illustrated with the student databases example.

Data reduction aims to decrease data volume while retaining the ability to yield similar or identical analytical outcomes. This is achieved through methods like dimensionality reduction, numerical summarization (encompassing mean, median, mode), data smoothing through binning methods, clustering, and feature subset selection.

Data transformation and discretization entail modifying the data's initial form to a format that enhances the predictive accuracy of machine learning models. Discretization, a subset of this process, converts continuous data into discrete variants.

Thank you

A special thank you to Qin Lv for her slides,
on which this lecture is based

Ravi Starzl, PhD - Data Mining Fall 2023