**CLASS-4 DATA PREPROCESSING PROJECT STEPS**

Let’s take a look at the established steps you’ll need to go through to make sure your data is successfully preprocessed.

1. Data quality assessment
2. Data cleaning
3. Data transformation
4. Data reduction
5. Feature Engineering
6. **Data quality assessment**

Take a good look at your data and get an idea of its overall quality, relevance to your project, and consistency. There are a number of data anomalies and inherent problems to look out for in almost any data set, for example:

* **Mismatched data types**: When you collect data from many different sources, it may come to you in different formats. While the ultimate goal of this entire process is to reformat your data for machines, you still need to begin with similarly formatted data. For example, if part of your analysis involves family income from multiple countries, you’ll have to convert each income amount into a single currency.
* **Mixed data values**: Perhaps different sources use different descriptors for features - for example, man or male. These value descriptors should all be made uniform.
* **Data outliers**: Outliers can have a huge impact on data analysis results. For example if you're averaging test scores for a class, and one student didn’t respond to any of the questions, their 0% could greatly skew the results.
* **Missing data**: Take a look for missing data fields, blank spaces in text, or unanswered survey questions. This could be due to human error or incomplete data. To take care of missing data, you’ll have to perform data cleaning.

1. **Data cleaning**

Data cleaning is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant data from a data set. Data cleaning is the most important step of preprocessing because it will ensure that your data is ready to go for your downstream needs.

Data cleaning will correct all of the inconsistent data you uncovered in your data quality assessment. Depending on the kind of data you’re working with, there are a number of possible cleaners you’ll need to run your data through.

Missing data

There are a number of ways to correct for missing data, but the two most

common are:

* **Ignore the tuples**: A tuple is an ordered list or sequence of numbers or entities. If multiple values are missing within tuples, you may simply discard the tuples with that missing information. This is only recommended for large data sets, when a few ignored tuples won’t harm further analysis.
* **Manually fill in missing data**: This can be tedious, but is definitely necessary when working with smaller data sets.

Noisy data

Data cleaning also includes fixing “noisy” data. This is data that includes unnecessary data points, irrelevant data, and data that’s more difficult to group

together.

* **Binning**: Binning sorts data of a wide data set into smaller groups of more similar data. It’s often used when analyzing demographics. Income, for example, could be grouped: $35,000-$50,000, $50,000-$75,000, etc.
* **Regression**: Regression is used to decide which variables will actually apply to your analysis. Regression analysis is used to smooth large amounts of data. This will help you get a handle on your data, so you’re not overburdened with unnecessary data.
* **Clustering**: Clustering algorithms are used to properly group data, so that it can be analyzed with like data. They’re generally used in unsupervised learning, when not a lot is known about the relationships within your data.

If you’re working with text data, for example, some things you should consider when cleaning your data are:

* Remove URLs, symbols, emojis, etc., that aren’t relevant to your analysis
* Translate all text into the language you’ll be working in
* Remove HTML tags
* Remove boilerplate email text
* Remove unnecessary blank text between words
* Remove duplicate data

After data cleaning, you may realize you have insufficient data for the task at hand. At this point you can also perform data wrangling or data enrichment to add new data sets and run them through quality assessment and cleaning again before adding them to your original data.

1. **Data transformation**

With data cleaning, we’ve already begun to modify our data, but data transformation will begin the process of turning the data into the proper format(s) you’ll need for analysis and other downstream processes.

This generally happens in one or more of the below:

1. Aggregation
2. Normalization
3. Feature selection
4. Discreditization
5. Concept hierarchy generation

* **Aggregation**: Data aggregation combines all of your data together in a uniform format.
* **Normalization**: Normalization scales your data into a regularized range so that you can compare it more accurately. For example, if you’re comparing employee loss or gain within a number of companies (some with just a dozen employees and some with 200+), you’ll have to scale them within a specified range, like -1.0 to 1.0 or 0.0 to 1.0.
* **Feature selection**: Feature selection is the process of deciding which variables (features, characteristics, categories, etc.) are most important to your analysis. These features will be used to train ML models. It’s important to remember, that the more features you choose to use, the longer the training process and, sometimes, the less accurate your results, because some feature characteristics may overlap or be less present in the data.
* **Discreditization**: Discreditization pools data into smaller intervals. It’s somewhat similar to binning, but usually happens after data has been cleaned. For example, when calculating average daily exercise, rather than using the exact minutes and seconds, you could join together data to fall into 0-15 minutes, 15-30, etc.
* **Concept hierarchy generation**: Concept hierarchy generation can add a hierarchy within and between your features that wasn’t present in the original data. If your analysis contains wolves and coyotes, for example, you could add the hierarchy for their genus: *canis*.

1. **Data reduction**

The more data you’re working with, the harder it will be to analyze, even after cleaning and transforming it. Depending on your task at hand, you may actually have more data than you need. Especially when working with text analysis, much of regular human speech is superfluous or irrelevant to the needs of the researcher. Data reduction not only makes the analysis easier and more accurate, but cuts down on data storage.

It will also help identify the most important features to the process at hand.

* **Attribute selection**: Similar to discreditization, attribute selection can fit your data into smaller pools. It, essentially, combines tags or features, so that tags like male/female and professor could be combined into male professor/female professor.
* **Numerosity reduction**: This will help with data storage and transmission. You can use a regression model, for example, to use only the data and variables that are relevant to your analysis.
* **Dimensionality reduction**: This, again, reduces the amount of data used to help facilitate analysis and downstream processes. Algorithms like Knearest neighbors use pattern recognition to combine similar data and make it more manageable.

1. **Feature Engineering**

**Feature Engineering**

Feature engineering is the process of using domain knowledge to extract features (characteristics, properties, attributes) from raw data. These features can be used to improve the performance of machine learning algorithms. Feature engineering is fundamental because the right features can facilitate a simpler model that performs better on new data. Here are some steps and techniques involved in feature engineering:

* **Feature Creation**: Creating new features based on existing ones or through domain knowledge. This can involve combining two variables to make a new one, breaking a variable into parts, or creating polynomial features from existing variables.
* **Feature Transformation**: Transforming features to enhance the relationship with the target variable. Common transformations include log transformation, square root transformation, and power transformation. These can help in dealing with skewed data or enhancing linear relationships.
* **Feature Encoding**: Converting categorical variables into a form that can be provided to ML algorithms to do a better job in prediction. Techniques include one-hot encoding, label encoding, and binary encoding. This step is crucial for handling non-numeric data.
* **Feature Selection**: Selecting the most significant features to train the model. This involves identifying and removing as much irrelevant and redundant information as possible. Techniques include backward elimination, forward selection, and using model-based methods like Lasso regression which includes feature selection inherently.
* **Feature Scaling**: Scaling features to a range or standard deviation. Standardization (scaling to a mean of 0 and a standard deviation of 1) and Min-Max normalization (scaling to a range between 0 and 1) are common methods. This is particularly important for models that are sensitive to the scale of data like SVM, k-nearest neighbors, and logistic regression.
* **Feature Extraction**: Reducing the number of features in a dataset by creating new features from the existing ones (dimensionality reduction), while still retaining the most important information. Techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and t-distributed Stochastic Neighbor Embedding (t-SNE).

Effective feature engineering requires an iterative process of creating, transforming, selecting, and evaluating features to find the best combination that improves the performance of machine learning models.