4/15/2024

Model Planning

- What are the strengths and weaknesses of the tools that you are using?
- What kinds of variables or models are they designed to do well?
- What packages are needed?

- Exploring the data to learn about the relationships between the variables
- Explore the individual variables
- Determine the types of models and which variables are the most useful
- Separate the data into test and training sets (maybe a third—does the model require only one test or multiple test sets? Cross-validation?)

This phase of data exploration goes beyond the original data exploration phase – now we have to prepare the data for a specific analysis, check our assumptions, etc.

Sometimes being removed from the domain of inquiry is beneficial, to be more objective; you might be less invested in a particular outcome; avoid gut feelings and pre-defined hunches. If these are offered, they must be tested against correlations with variables.

The flip side is also true, and can be a risk.

Types of models, options that are available:

- Map/Reduce
- Natural Language Processing (NLP)
- Clustering
- Classification
- Regression
- Graph Theory

Don’t limit yourself to just one model – chose several to analyze and select one that produces the best predictive value.

Take care that our test/train split doesn’t contaminate each other.

- Normalize the data after doing the test/train split, not before
- Do dimensionality reduction after the test/train split, not before

Map/Reduce
Is a big data approach (Hadoop, Hive, Spark), take data and map it into multiple processors, and them recombines the results into a single output

Spark works similarly to Hadoop (Python works with Spark using pyspark), Spark processes data in memory while Hadoop stores to disk – Spark is faster

Hadoop systems may be less expensive and are suitable for analyses that can run overnight. Spark is better for more real-time processing.

Hadoop is better for linear processing. Spark is better for iterative processing, graphs and joining data sets.

Natural Language Processing
Analysis of language in text or speech form

NLP is challenging because it is messy

Grew out of linguistics, semantics and syntax employed mathematical frameworks, formal methods, but many interesting problems remain unsolved

Modern NLP grew out of computational linguistics, developed a more statistical approach

NLTK is the Python package that does NLP

Clustering Algorithms
Often used as a type of unsupervised learning—but can be modified to do as supervised or semi-supervised

Density-based methods—DBSCAN, OPTICS, etc.
Hierarchical methods –
    Agglomerative methods (bottom up)
    Divisive (top-down)
CURE, BIRCH, etc.

Partitioning methods – k-means, CLARANS, LDA, etc.
Grid-based methods – STING, CLIQUE, etc.

k-means is the most common of these methods.

Resources:

1. https://stevetodd.typepad.com/my_weblog/2012/05/phase-3-innovation-analytics-.html
2. https://www.sciencedirect.com/topics/computer-science/mapreduce-model
3. https://www.scnsoft.com/blog/spark-vs-hadoop-mapreduce
4. https://machinelearningmastery.com/natural-language-processing/
5. https://monkeylearn.com/sentiment-analysis/
6. https://www.geeksforgeeks.org/clustering-in-machine-learning/
7. https://www.edureka.co/blog/classification-in-machine-learning/
8. http://www.sthda.com/english/wiki/regression-analysis-essentials-for-machine-learning
9. https://towardsdatascience.com/graph-theory-and-deep-learning-know-hows-6556b0e9891b
10. https://machinelearningmastery.com/statistical-hypothesis-tests/

Extended commentary:

Model planning is an important stage in data analysis where the problem to be solved is defined and a plan is developed to achieve the solution. The goal is to establish a clear understanding of the project and to identify the steps required to reach the desired outcome.

The process of model planning involves several key steps, including:

Defining the problem: This involves identifying the business problem or question that the model will address.

Identifying the data requirements: This involves identifying the data required to build the model, including the data sources, data quality requirements, and data collection methods.

Identifying the modeling techniques: This involves selecting the appropriate modeling techniques based on the problem and data requirements.

Developing a data model: This involves creating a detailed data model that describes the data, the relationships between the data, and the analytical models that will be used to analyze the data.

Developing a project plan: This involves developing a detailed project plan that outlines the steps required to build the model, the timelines for each step, and the resources required.

Defining success metrics: This involves establishing metrics to evaluate the success of the project, such as accuracy, precision, recall, and F1 score.

Overall, model planning is a critical component of the data analysis process as it provides a roadmap for the project and helps ensure that the project is successful.

There are various types of models available for data analysis, and the choice of model will depend on the specific problem and the type of data being analyzed. Some common types of models include:

Linear regression models: These models are used to analyze the relationship between two or more variables, with the aim of predicting the value of one variable based on the values of the others.

Time series models: These models are used to analyze time-dependent data, such as stock prices or weather patterns, with the aim of predicting future values.

Classification models: These models are used to classify data into predefined categories, based on the values of certain variables.

Clustering models: These models are used to group data into clusters or segments, based on similarities between the variables.

Neural networks: These models are inspired by the structure of the human brain and can be used for a wide range of applications, including image and speech recognition, natural language processing, and predictive analytics.

Decision trees: These models are used to represent decisions and their possible consequences in a tree-like structure, with the aim of identifying the best course of action based on the available data.

Support vector machines: These models are used for classification and regression analysis, and are particularly useful when working with complex datasets with multiple variables.

These are just a few examples of the many types of models available for data analysis, and the choice of model will depend on the specific problem and the type of data being analyzed.

When performing a test/train split for a machine learning model, some common errors include:

Not randomly shuffling the data before splitting: If the data is not randomly shuffled, the training set and test set may not be representative of the entire dataset, leading to biased results.

Using too small of a sample size for the test set: If the test set is too small, the evaluation metrics may not accurately represent the model's performance on new, unseen data.

Data leakage: This occurs when information from the test set is inadvertently used during training, leading to overly optimistic evaluation metrics. This can happen, for example, when data is normalized before the split, or when the same feature engineering techniques are used on both the training and test sets.

Not stratifying the split: If the dataset is imbalanced, meaning there are many more examples of one class than another, it's important to stratify the split to ensure that both the training and test sets contain representative examples of each class.

Ignoring the temporal aspect of the data: If the data has a temporal aspect, such as stock prices or weather data, it's important to split the data chronologically so that the model is tested on data that comes after the training period.

Distributed computing environments can significantly impact data analysis by allowing for parallel processing of large datasets. With distributed computing, a single task can be broken down into smaller sub-tasks that are distributed among multiple machines, which can significantly reduce the time it takes to process the data.

Distributed computing environments can also help with fault tolerance, as multiple machines can work together to complete a task. If one machine fails or crashes, the remaining machines can pick up the slack and continue the processing without interruption.

One common distributed computing environment used for data analysis is Apache Hadoop, which is an open-source framework that allows for the distributed processing of large datasets across clusters of computers. Other examples include Apache Spark and Apache Storm.

However, distributed computing environments also come with some challenges. These include the need for specialized hardware and software, as well as the need for a skilled team to manage and maintain the environment. Additionally, the distributed nature of the environment can make debugging and troubleshooting more difficult.

MapReduce is a programming model and an associated implementation for processing and generating large data sets in a distributed computing environment. It allows parallel processing of data across large clusters of computers.

The MapReduce algorithm works by breaking down a large data processing task into smaller sub-tasks, which can be processed in parallel across many different computers. The process is split into two phases:

Map phase: In this phase, the input data is divided into smaller subsets and distributed to multiple machines for processing. Each machine processes the subset of data independently and generates a key-value pair as output.

Reduce phase: In this phase, the key-value pairs produced in the map phase are combined and aggregated to produce the final output.

The MapReduce algorithm is fault-tolerant and handles failures by automatically reassigning tasks to other machines in the cluster. The data is stored in a distributed file system, such as Hadoop Distributed File System (HDFS), which allows for scalability and fault tolerance.

MapReduce is widely used for large-scale data processing tasks, such as analyzing web logs, processing large-scale image and video data, and performing data mining and machine learning tasks.

Natural Language Processing (NLP) is a subfield of computer science and artificial intelligence (AI) concerned with the interaction between computers and human languages. It involves developing algorithms and computational models that can understand, interpret, and generate human language. NLP technologies are used in a wide range of applications, including speech recognition, sentiment analysis, machine translation, chatbots, and virtual assistants.

NLP involves several key tasks, including:

Tokenization: Breaking text into individual words or tokens.

Part-of-speech (POS) tagging: Identifying the grammatical parts of speech of words in a sentence.

Named entity recognition (NER): Identifying and classifying named entities such as people, organizations, and locations.

Sentiment analysis: Identifying the sentiment or emotion expressed in a piece of text.

Topic modeling: Identifying the main topics or themes in a collection of documents.

NLP algorithms use a combination of statistical and rule-based approaches to analyze and understand natural language. Deep learning methods such as neural networks have also been increasingly used in NLP tasks in recent years, leading to significant improvements in accuracy and performance.

Clustering is a technique in unsupervised learning used to group similar data points together based on the similarity of their features. Clustering algorithms aim to identify meaningful groups or clusters within a dataset, without prior knowledge of the group membership.

There are various clustering algorithms, each with its strengths and weaknesses, and they can be broadly classified into two categories: hierarchical clustering and partitioning clustering.

Hierarchical clustering is a method that creates a tree-like structure of clusters, starting with all data points as individual clusters and gradually combining them into larger clusters based on their similarity until a single cluster that contains all data points is formed. The two main types of hierarchical clustering are agglomerative clustering and divisive clustering.

Partitioning clustering is a method that partitions the dataset into a fixed number of clusters, where each data point belongs to only one cluster. The popular partitioning clustering algorithms are k-means, k-modes, and DBSCAN.

K-means is a popular clustering algorithm that partitions data points into k number of clusters based on their similarities. The algorithm randomly selects k initial centroids, computes the distance of each data point from each centroid, and assigns each point to the nearest centroid. The algorithm then updates the centroids based on the newly formed clusters and iterates the process until the centroids no longer change or a maximum number of iterations is reached.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is another popular clustering algorithm that groups data points based on their density. The algorithm identifies dense regions of data points and considers them as clusters, while the less dense regions are considered noise. DBSCAN is robust to outliers and can discover clusters of any shape and size.

Clustering algorithms are widely used in various fields, such as customer segmentation, image processing, anomaly detection, and recommendation systems.