SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107
AN AUTONOMOUS INSTITUTION
Accredited by NBA–AICTE and Accredited by NAAC–UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

COURSE NAME : 19CS407- DATA ANALYTICS WITH R

II YEAR / IV SEMESTER

PRESCRIPTIVE ANALYTICS

Introduction

Prescriptive analytics is an advanced form of data analytics that focuses on determining the best course of action or decision to take in a given situation. Unlike descriptive analytics (which focuses on understanding what has happened) and predictive analytics (which focuses on forecasting what might happen), prescriptive analytics goes a step further by providing recommendations on what actions to take to optimize outcomes.

Prescriptive analytics utilizes a combination of historical data, real-time data, statistical analysis, mathematical modeling, and optimization techniques to generate actionable insights. It considers various constraints, objectives, and potential outcomes to suggest the most favorable decision or action to achieve a specific goal.

The process of prescriptive analytics involves several steps:

Data Collection: Gathering relevant and accurate data from various sources, including historical data, real-time data feeds, and external data.

Data Analysis: Applying advanced analytical techniques such as statistical analysis, machine learning, and data mining to extract meaningful insights from the collected data.

Model Development: Creating mathematical models and algorithms that represent the problem domain, incorporating the relevant variables, constraints, and objectives.

Simulation and Optimization: Running simulations and optimization algorithms to evaluate different scenarios and identify the best possible outcomes based on predefined goals and constraints.

Decision Support: Presenting the recommended actions or decisions to the decision-makers through visualizations, reports, dashboards, or interactive tools.

Prescriptive analytics can be applied to various domains and industries, including supply chain management, finance, healthcare, marketing, logistics, and manufacturing. It helps organizations make informed decisions, optimize operations, minimize risks, increase efficiency, and gain a competitive advantage.

However, it's important to note that prescriptive analytics relies heavily on the quality of data, the accuracy of models, and the assumptions made during the analysis. Additionally, the implementation of prescriptive analytics requires careful consideration of ethical, legal, and social implications, as it involves making decisions that impact individuals, organizations, and society as a whole.

Types

here are different types or approaches to prescriptive analytics, depending on the specific problem or decision-making context. Here are some common types of prescriptive analytics techniques:

Optimization Models: Optimization models use mathematical programming techniques to find the best solution from a set of possible options. These models consider constraints, objectives, and decision variables to optimize outcomes. Linear programming, integer programming, and nonlinear programming are commonly used optimization techniques.

Decision Trees: Decision trees are graphical models that represent decisions and their possible consequences in a tree-like structure. They use statistical and probabilistic techniques to determine the optimal decision path based on various factors and outcomes.

Simulation: Simulation involves creating a computer-based model that mimics real-world scenarios and allows for experimentation and analysis. It helps decision-makers understand the potential impacts of different choices and optimize decision-making under uncertain conditions.

Rule-Based Systems: Rule-based systems utilize if-then rules or logical statements to provide recommendations or decisions. These rules are based on expert knowledge or predefined heuristics. Rule-based systems are commonly used in areas such as expert systems and knowledge management.

Prescriptive Machine Learning: Prescriptive machine learning combines elements of predictive analytics and optimization to provide recommendations for decision-making. It involves training machine learning models on historical data and using them to predict outcomes and optimize decisions.

Genetic Algorithms: Genetic algorithms are optimization techniques inspired by the process of natural selection. They involve generating a population of potential solutions, evaluating their fitness based on defined criteria, and iteratively evolving the population to find the best solution.

Multi-Criteria Decision Analysis (MCDA): MCDA is an approach that considers multiple criteria and objectives when making decisions. It involves evaluating options based on various criteria, assigning weights to each criterion, and aggregating the results to identify the best alternative.

These are just a few examples of the types of techniques used in prescriptive analytics. The choice of technique depends on the nature of the problem, available data, computational resources, and specific requirements of the decision-making process.

Example

Let's consider a supply chain management example to illustrate how prescriptive analytics can be applied:

Suppose a retail company wants to optimize its inventory levels across its network of stores to minimize costs while ensuring product availability. The company has historical sales data, supplier information, transportation costs, and storage costs. The goal is to determine the optimal distribution of inventory among the stores.

Here's how prescriptive analytics can be applied in this scenario:

Data Collection: The company gathers historical sales data, supplier information, transportation costs, and storage costs for each store in its network.

Data Analysis: Advanced analytics techniques are applied to analyze the collected data and identify patterns, demand trends, and correlations between variables.

Model Development: An optimization model is developed that considers factors such as demand patterns, lead times, transportation costs, storage capacities, and service level objectives. The model includes decision variables (e.g., how much inventory to allocate to each store) and constraints (e.g., maximum storage capacity, minimum service level).

Simulation and Optimization: The model is used to simulate different scenarios and optimize the allocation of inventory among the stores. The simulation takes into account factors like seasonality, demand fluctuations, and supplier reliability. Optimization algorithms are applied to find the combination of inventory levels that minimizes costs while meeting service level targets.

Decision Support: The results of the optimization process are presented to the decision-makers through visualizations and reports. They can see the recommended inventory allocation for each store, along with associated costs and projected service levels. Decision-makers can analyze the trade-offs and make informed decisions based on the insights provided.

By leveraging prescriptive analytics in this supply chain management example, the retail company can optimize its inventory distribution strategy, reduce costs associated with excess inventory or stockouts, and improve customer satisfaction by ensuring product availability.

It's worth noting that this is a simplified example, and in practice, prescriptive analytics models can be much more complex, incorporating additional factors and constraints based on the specific requirements of the problem at hand.

Creating data for analytics through designed experiments

Creating data for analytics through designed experiments is a systematic approach to gather data by intentionally manipulating variables and measuring their effects on the outcome of interest. It involves planning and executing controlled experiments to generate data that can be analyzed to gain insights and make informed decisions.

Here are the key steps involved in creating data through designed experiments:

Define the Research Objective: Clearly articulate the research question or objective that you want to address through the experiment. This could be testing the effectiveness of a new product feature, optimizing a manufacturing process, or understanding the impact of a marketing campaign.

Identify Variables: Determine the independent variables (factors) that you want to investigate and the dependent variable (response) that you want to measure. Independent variables are the factors you control and manipulate, while the dependent variable is the outcome you observe and measure.

Design the Experiment: Select an appropriate experimental design based on the research objective and the number of variables involved. Common designs include factorial designs, randomized controlled trials, and response surface designs. Consider factors such as the number of levels for each variable, replication, randomization, and blocking.

Determine Sample Size: Determine the number of experimental units (observations) needed to achieve statistical power and precision in your analysis. Consider factors such as desired level of significance, effect size, and variability in the data.

Conduct the Experiment: Implement the experimental design by manipulating the independent variables and collecting data on the dependent variable. Ensure proper control and randomization to minimize bias and confounding effects.

Analyze the Data: Use statistical analysis techniques to analyze the collected data and draw conclusions. This may involve techniques such as analysis of variance (ANOVA), regression analysis, or other appropriate statistical models. Determine the significance of the independent variables and their impact on the dependent variable.

Interpret and Draw Insights: Interpret the results of the analysis and draw insights based on the observed effects of the variables. Understand the relationships and interactions between the factors and their influence on the outcome. Use the insights to make data-driven decisions and recommendations.

By intentionally designing experiments to create data, organizations can gain a deeper understanding of the relationships between variables and their impact on outcomes. This approach allows for more precise and controlled analysis, enabling better decision-making and optimization of processes, products, and strategies.

Let's consider an example to illustrate how designed experiments can be used to create data for analytics:

Suppose a company wants to determine the optimal packaging design for a new product to maximize customer satisfaction and minimize product damage during transportation. The company decides to conduct a designed experiment to gather data and make informed decisions about the packaging design.

Here's how the process could unfold:

Define the Research Objective: The research objective is to identify the packaging design that minimizes product damage and maximizes customer satisfaction during transportation.

Identify Variables: The independent variables (factors) could include the type of packaging material (A, B, C), the cushioning material (X, Y, Z), and the sealing method (Tape, Glue). The dependent variables (response) could include measures of product damage and customer satisfaction.

Design the Experiment: The company decides to use a factorial design to investigate the main effects and potential interactions between the variables. They choose a 3x3x2 factorial design, resulting in 18 unique combinations of the independent variables. The design is randomized to minimize the effects of confounding variables.

Determine Sample Size: The company determines the sample size based on factors such as desired level of significance, effect size, and variability. They decide to produce and test 10 units for each combination, resulting in a total of 180 experimental units.

Conduct the Experiment: The company manufactures the products using each combination of packaging design factors. They carefully package and label each unit according to the experimental design. The units are then transported using standard shipping methods.

Analyze the Data: After transportation, the company inspects each unit for damage and collects customer satisfaction feedback through surveys or other means. They record the level of product damage and customer satisfaction scores for each combination of packaging design factors.

Interpret and Draw Insights: The company performs statistical analysis on the collected data, such as conducting an ANOVA to determine the significance of the packaging design factors and their interactions. They analyze the data to understand the effects of different packaging design combinations on product damage and customer satisfaction.

Based on the analysis, the company may find that certain combinations of packaging materials, cushioning materials, and sealing methods result in significantly lower product damage and higher customer satisfaction. They can use these insights to make data-driven decisions about the optimal packaging design for their product, potentially leading to cost savings, improved customer experience, and reduced product losses during transportation.

This example demonstrates how designed experiments can be used to systematically create data and generate insights that guide decision-making in various areas such as product development, manufacturing, and quality control.

Creating data for analytics through active learning

Creating data for analytics through active learning is a process that involves iteratively selecting and labeling data instances to improve the performance of a machine learning model. It aims to optimize the use of limited labeled data by strategically choosing the most informative samples for annotation, reducing the need for extensive labeling efforts.

Here's an overview of the process of creating data through active learning:

Initial Model Training: Start with a small labeled dataset to train an initial machine learning model. This initial model serves as a baseline for active learning.

Model Prediction: Use the trained model to make predictions on a larger pool of unlabeled data. These predictions are used to estimate the uncertainty or in formativeness of each unlabeled instance.

Query Strategy: Apply a query strategy to select the most informative or uncertain instances from the pool of unlabeled data for annotation. The query strategy can be based on various approaches such as uncertainty sampling, diversity sampling, or expected model change.

Human Annotation: The selected instances are sent for human annotation or labeling. An expert or a team of annotators assign the correct labels to these instances.

Model Retraining: Incorporate the newly labeled data into the training set and retrain the machine learning model using both the existing labeled data and the newly labeled instances.

Iterative Process: Repeat steps 2 to 5 iteratively, continually selecting and labeling the most informative instances and updating the model. The goal is to improve the model's performance with each iteration while minimizing the number of labeled instances required.

Stopping Criteria: Determine stopping criteria for the active learning process. This could be a predefined number of iterations, a performance threshold, or a budget constraint on the number of labeled instances.

By leveraging active learning, organizations can optimize the use of their resources and reduce the time and effort required for manual labeling. It enables more efficient and targeted data collection, especially when labeling large datasets is time-consuming or expensive.

Active learning is particularly useful in scenarios where there is a scarcity of labeled data or when the cost of annotation is high. It can be applied in various domains such as text classification, image recognition, fraud detection, and recommendation systems.

Note that the choice of query strategy and the effectiveness of active learning depend on the specific problem, the characteristics of the dataset, and the machine learning algorithms used. Experimentation and fine-tuning are often necessary to determine the most effective active learning approach for a given task.

Let's consider an example problem where active learning can be applied: sentiment analysis for customer reviews.

Suppose a company wants to analyze customer sentiment towards its products by classifying customer reviews as positive, negative, or neutral. However, manually labeling a large dataset of customer reviews for sentiment analysis would be time-consuming and costly. Active learning can help optimize the labeling process by selecting the most informative reviews for annotation.

Here's how the process could unfold:

Initial Model Training: Train an initial sentiment analysis model using a small labeled dataset of customer reviews. The model is capable of classifying reviews as positive, negative, or neutral.

Model Prediction: Apply the trained model to a larger pool of unlabeled customer reviews. The model assigns sentiment labels to each review based on its current understanding.

Query Strategy: Use an active learning query strategy, such as uncertainty sampling. This strategy selects reviews for annotation that the model is most uncertain about or considers the most informative.

Human Annotation: Send the selected reviews for annotation to human annotators who assign the correct sentiment labels (positive, negative, or neutral) to each review.

Model Retraining: Incorporate the newly labeled reviews into the training set and retrain the sentiment analysis model using both the existing labeled data and the newly labeled reviews.

Iterative Process: Repeat steps 2 to 5 iteratively. In each iteration, the model makes predictions on the unlabeled reviews, selects the most informative or uncertain ones for annotation, and incorporates the labeled data to improve its performance.

Stopping Criteria: Set stopping criteria, such as reaching a desired accuracy level or a predefined number of iterations, to determine when to stop the active learning process.

Through the iterative process of active learning, the sentiment analysis model learns from the most informative reviews, gradually improving its performance with fewer labeled reviews compared to traditional approaches. This approach significantly reduces the effort and cost associated with manually labeling a large dataset.

Active learning allows the company to focus human annotation efforts on the most critical reviews, such as those with ambiguous sentiment or high uncertainty, leading to a more accurate sentiment analysis model. The improved model can then be used to analyze sentiment on a larger scale and gain valuable insights into customer perceptions of their products.

Note that the specific active learning query strategy and the details of the sentiment analysis model would depend on the specific requirements and available resources of the company.

Creating data for analytics through Reinforcement learning

Creating data for analytics through reinforcement learning is a process where an agent interacts with an environment to learn optimal actions by trial and error. In this context, data is generated as the agent explores and learns from its interactions with the environment.

Here's an overview of the process of creating data through reinforcement learning:

Define the Environment: Specify the environment in which the agent operates. The environment includes states, actions, rewards, and a transition function that describes how the state evolves based on actions taken by the agent.

Initialize the Agent: Set up the reinforcement learning agent, which typically includes an algorithm (e.g., Q-learning, Deep Q-Networks) and a representation (e.g., neural network). The agent's objective is to learn a policy that maximizes cumulative rewards.

Interaction Loop: The agent interacts with the environment by observing the current state, taking an action, receiving a reward, and transitioning to the next state. This loop continues iteratively.

Exploration vs. Exploitation: During the interaction loop, the agent balances exploration (trying new actions to gather information about the environment) and exploitation (leveraging learned knowledge to take actions that yield high rewards).

Data Generation: As the agent interacts with the environment, it generates data consisting of state-action pairs, rewards, and the resulting next states. This data serves as a learning dataset for the agent.

Update Agent's Policy: The agent uses the generated data to update its policy (i.e., the strategy for selecting actions) using the chosen reinforcement learning algorithm. The objective is to improve the policy based on observed rewards and experiences.

Iterate and Refine: Repeat the interaction loop and update process, allowing the agent to explore the environment, gather new data, and continuously improve its policy over time.

By leveraging reinforcement learning, organizations can train agents to learn optimal strategies and actions in complex and dynamic environments. This approach is particularly useful when the environment is not fully known or analytically solvable, and trial-and-error learning is required to discover effective solutions.

Reinforcement learning has been successfully applied in various domains, including robotics, gaming, finance, healthcare, and supply chain management. It allows for data-driven decision-making by creating data through the agent's interactions with the environment.

It's important to note that the performance and effectiveness of reinforcement learning depend on factors such as the design of the environment, the choice of algorithms, hyper parameter tuning, and the availability of computational resources

Let's consider an example of creating data for analytics through reinforcement learning in the context of training an autonomous driving agent.

Suppose a company wants to develop an autonomous driving system that can navigate a complex urban environment while obeying traffic rules and ensuring passenger safety. The company decides to use reinforcement learning to train an agent to learn optimal driving policies.

Here's how the process could unfold:

Define the Environment: Create a simulated urban environment that includes roads, intersections, traffic lights, pedestrians, and other vehicles. Specify the states (e.g., location, speed, surrounding objects), actions (e.g., accelerate, brake, steer), and rewards (e.g., reaching the destination, avoiding collisions).

Initialize the Agent: Set up the reinforcement learning agent with an appropriate algorithm, such as Deep Q-Networks (DQN). The agent's objective is to learn a policy that maximizes the cumulative reward, which includes reaching the destination safely and efficiently.

Interaction Loop: The agent interacts with the simulated environment by observing the current state (e.g., sensor data), selecting an action, receiving a reward based on the action taken, and transitioning to the next state. This loop continues iteratively.

Exploration vs. Exploitation: The agent balances exploration and exploitation. Initially, it explores the environment by taking random or uncertain actions to gather data and learn about the consequences. As training progresses, the agent gradually shifts towards exploitation, leveraging its learned knowledge to make more informed and optimal driving decisions.

Data Generation: As the agent interacts with the environment, it generates data consisting of state-action pairs, rewards, and the resulting next states. This data serves as a learning dataset for the agent.

Update Agent's Policy: The agent uses the generated data to update its policy using the chosen reinforcement learning algorithm (e.g., DQN). The agent learns from the rewards received and experiences gained during its interactions with the environment, improving its driving policy over time.

Iterate and Refine: Repeat the interaction loop and update process, allowing the agent to explore the simulated environment, gather new data, and continuously refine its driving policy.

Through reinforcement learning, the autonomous driving agent learns to navigate the complex urban environment, adapt to traffic conditions, follow traffic rules, avoid collisions, and reach the destination efficiently. The process of creating data through the agent's interactions with the environment enables the agent to learn from trial and error and make data-driven decisions for safe and effective driving.

It's important to note that developing an autonomous driving system involves many considerations beyond just reinforcement learning, such as perception systems, control algorithms, and safety mechanisms. Reinforcement learning is one component that helps train the agent's decision-making capabilities in a complex environment.