20 Core
Data Science
Concepts

1_ Dataset

Just as the name implies, data science is a branch of science that applies the scientific method to data with the goal of studying the relationships between different features and drawing out meaningful conclusions based on these relationships. Data is, therefore, the key component in data science. A dataset is a particular instance of data that is used for analysis or model building at any given time. A dataset comes in different flavors such as numerical data, categorical data, text data, image data, voice data, and video data. A dataset could be static (not changing) or dynamic (changes with time, for example, stock prices). Moreover, a dataset could depend on space as well. For example, temperature data in the United States would differ significantly from temperature data in Africa. For beginning data science projects, the most popular type of dataset is a dataset containing numerical data that is typically stored in a comma-separated values (CSV) file format.

2_ Data Wrangling

Data wrangling is the process of converting data from its raw form to a tidy form ready for analysis. Data wrangling is an important step in data preprocessing and includes several processes like data importing, data cleaning, data structuring, string processing, HTML parsing, handling dates and times, handling missing data, and text mining.

The process of data wrangling is a critical step for any data scientist. Very rarely is data easily accessible in a data science project for analysis. It is more likely for the data to be in a file, a database, or extracted from documents such as web pages, tweets, or PDFs. Knowing how to wrangle and clean data will enable you to derive critical insights from your data that would otherwise be hidden.

3_ Data Visualization

Data Visualization is one of the most important branches of data science. It is one of the main tools used to analyze and study relationships between different variables. Data visualization (e.g., scatter plots, line graphs, bar plots, histograms, qqplots, smooth densities, boxplots, pair plots, heat maps, etc.) can be used for descriptive analytics. Data visualization is also used in machine learning for data preprocessing and analysis, feature selection, model building, model testing, and model evaluation. When preparing a data visualization, keep in mind that data visualization is more of an Art than Science. To produce a good visualization, you need to put several pieces of code together for an excellent end result.

4_ Outliers

An outlier is a data point that is very different from the rest of the dataset. Outliers are often just bad data, e.g., due to a malfunctioned sensor; contaminated experiments; or human error in recording data. Sometimes, outliers could indicate something real such as a malfunction in a system. Outliers are very common and are expected in large datasets. One common way to detect outliers in a dataset is by using a box plot.

Outliers can significantly degrade the predictive power of a machine learning model. A common way to deal with outliers is to simply omit the data points. However, removing real data outliers can be too optimistic, leading to non-realistic models. Advanced methods for dealing with outliers include the RANSAC method.

5_ Data Imputation

Most datasets contain missing values. The easiest way to deal with missing data is simply to throw away the data point. However, the removal of samples or dropping of entire feature columns is simply not feasible because we might lose too much valuable data. In this case, we can use different interpolation techniques to estimate the missing values from the other training samples in our dataset. One of the most common interpolation techniques is mean imputation, where we simply replace the missing value with the mean value of the entire feature column. Other options for imputing missing values are median or most frequent (mode), where the latter replaces the missing values with the most frequent values. Whatever imputation method you employ in your model, you have to keep in mind that imputation is only an approximation, and hence can produce an error in the final model. If the data supplied was already preprocessed, you would have to find out how missing values were considered. What percentage of the original data was discarded? What imputation method was used to estimate missing values?

6_ Data Scaling

**Scaling your features will help improve the quality and predictive power of your model.
In order to bring features to the same scale, we could decide to use either normalization or standardization of features. Most often, we assume data is normally distributed and default towards standardization, but that is not always the case. It is important that before deciding whether to use either standardization or normalization, you first take a look at how your features are statistically distributed. If the feature tends to be uniformly distributed, then we may use normalization (MinMaxScaler). If the feature is approximately Gaussian, then we can use standardization (StandardScaler). Again, note that whether you employ normalization or standardization, these are also approximative methods and are bound to contribute to the overall error of the model.**

7_ Principal Component Analysis (PCA)

Large datasets with hundreds or thousands of features often lead to redundancy especially when features are correlated with each other. Training a model on a high-dimensional dataset having too many features can sometimes lead to overfitting (the model captures both real and random effects). In addition, an overly complex model having too many features can be hard to interpret. One way to solve the problem of redundancy is via feature selection and dimensionality reduction techniques such as PCA. Principal Component Analysis (PCA) is a statistical method that is used for feature extraction. PCA is used for high-dimensional and correlated data. The basic idea of PCA is to transform the original space of features into the space of the principal component. A PCA transformation achieves the following:

a) Reduce the number of features to be used in the final model by focusing only on the components accounting for the majority of the variance in the dataset.

b) Removes the correlation between features.

8_ Linear Discriminant Analysis (LDA)

PCA and LDA are two data preprocessing linear transformation techniques that are often used for dimensionality reduction to select relevant features that can be used in the final machine learning algorithm. PCA is an unsupervised algorithm that is used for feature extraction in high-dimensional and correlated data. PCA achieves dimensionality reduction by transforming features into orthogonal component axes of maximum variance in a dataset. The goal of LDA is to find the feature subspace that optimizes class separability and reduce dimensionality. Hence, LDA is a supervised algorithm. An in-depth description of PCA and LDA can be found in this book: Python Machine Learning by Sebastian Raschka, Chapter 5.

9. Data Partitioning

In machine learning, the dataset is often partitioned into training and testing sets. The model is trained on the training dataset and then tested on the testing dataset. The testing dataset thus acts as the unseen dataset, which can be used to estimate a generalization error (the error expected when the model is applied to a real-world dataset after the model has been deployed).

10_ Supervised Learning

These are machine learning algorithms that perform learning by studying the relationship between the feature variables and the known target variable. Supervised learning has two subcategories:

a) Continuous Target Variables

Algorithms for predicting continuous target variables include Linear Regression, KNeighbors regression (KNR), and Support Vector Regression (SVR).

b) Discrete Target Variables

Algorithms for predicting discrete target variables include:

- Perceptron classifier
- Logistic Regression classifier
- Support Vector Machines (SVM)
- Decision tree classifier
- K-nearest classifier
- Naive Bayes classifier

11_ Unsupervised Learning

In unsupervised learning, we are dealing with unlabeled data or data of unknown structure. Using unsupervised learning techniques, we are able to explore the structure of our data to extract meaningful information without the guidance of a known outcome variable or reward function. K-means clustering is an example of an unsupervised learning algorithm.

12_ Reinforcement Learning

In reinforcement learning, the goal is to develop a system (agent) that improves its performance based on interactions with the environment. Since the information about the current state of the environment typically also includes a so-called reward signal, we can think of reinforcement learning as a field related to supervised learning. However, in reinforcement learning, this feedback is not the correct ground truth label or value but a measure of how well the action was measured by a reward function. Through the interaction with the environment, an agent can then use reinforcement learning to learn a series of actions that maximize this reward.

13_ Model Parameters and Hyperparameters

In a machine learning model, there are two types of parameters:

a) Model Parameters: These are the parameters in the model that must be determined using the training data set. These are the fitted parameters.

b) Hyperparameters: These are adjustable parameters that must be tuned to obtain a model with optimal performance.

It is important that during training, the hyperparameters be tuned to obtain the model with the best performance (with the best-fitted parameters).

14_ Cross-validation

Cross-validation is a method of evaluating a machine learning model's performance across random samples of the dataset. This assures that any biases in the dataset are captured. Cross-validation can help us to obtain reliable estimates of the model's generalization error, that is, how well the model performs on unseen data.

In k-fold cross-validation, the dataset is randomly partitioned into training and testing sets. The model is trained on the training set and evaluated on the testing set. The process is repeated k-times. The average training and testing scores are then calculated by averaging over the k-folds.

15_ Bias-variance Tradeoff

In statistics and machine learning, the bias-variance tradeoff is the property of a set of predictive models whereby models with a lower bias in parameter estimation have a higher variance of the parameter estimates across samples and vice versa. The bias-variance dilemma or problem is the conflict in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:

The bias is an error from erroneous assumptions in the learning algorithm. High bias (overly simple) can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The variance is an error from sensitivity to small fluctuations in the training set. High variance (overly complex) can cause an algorithm to model the random noise in the training data rather than the intended outputs (overfitting).

It is important to find the right balance between model simplicity and complexity.

16_ Evaluation Metrics

In machine learning (predictive analytics), there are several metrics that can be used for model evaluation. For example, a supervised learning (continuous target) model can be evaluated using metrics such as the R2 score, mean square error (MSE), or mean absolute error (MAE). Furthermore, a supervised learning (discrete target) model, also referred to as a classification model, can be evaluated using metrics such as accuracy, precision, recall, f1 score, and the area under ROC curve (AUC).

17_ Uncertainty Quantification

It is important to build machine learning models that will yield unbiased estimates of uncertainties in calculated outcomes. Due to the inherent randomness in the dataset and model, evaluation parameters such as the R2 score are random variables, and thus it is important to estimate the degree of uncertainty in the model.

18_ Math Concepts

a) Basic Calculus: Most machine learning models are built with a dataset having several features or predictors. Hence, familiarity with multivariable calculus is extremely important for building a machine learning model. Here are the topics you need to be familiar with:

Functions of several variables; Derivatives and gradients; Step function, Sigmoid function, Logit function, ReLU (Rectified Linear Unit) function; Cost function; Plotting of functions; Minimum and Maximum values of a function

Math Concepts

b) Basic Linear Algebra: Linear algebra is the most important math skill in machine learning. A data set is represented as a matrix. Linear algebra is used in data preprocessing, data transformation, dimensionality reduction, and model evaluation. Here are the topics you need to be familiar with:

Vectors; Norm of a vector; Matrices; Transpose of a matrix; The inverse of a matrix; The determinant of a matrix; Trace of a Matrix; Dot product; Eigenvalues; Eigenvectors

Math Concepts

c) Optimization Methods: Most machine learning algorithms perform predictive modeling by minimizing an objective function, thereby learning the weights that must be applied to the testing data in order to obtain the predicted labels. Here are the topics you need to be familiar with:

Cost function/Objective function; Likelihood function; Error function; Gradient Descent Algorithm and its variants (e.g., Stochastic Gradient Descent Algorithm)

19_ Statistics and Probability Concepts

Statistics and Probability are used for visualization of features, data preprocessing, feature transformation, data imputation, dimensionality reduction, feature engineering, model evaluation, etc. Here are the topics you need to be familiar with:

Mean, Median, Mode, Standard deviation/variance, Correlation coefficient and the covariance matrix, Probability distributions (Binomial, Poisson, Normal), p-value, Bayes Theorem (Precision, Recall, Positive Predictive Value, Negative Predictive Value, Confusion Matrix, ROC Curve), Central Limit Theorem, R_2 score, Mean Square Error (MSE), A/B Testing, Monte Carlo Simulation

20_ Productivity Tools

A typical data analysis project may involve several parts, each including several data files and different scripts with code. Keeping all these organized can be challenging. Productivity tools help you to keep projects organized and to maintain a record of your completed projects. Some essential productivity tools for practicing data scientists include tools such as Unix/Linux, git and GitHub, RStudio, and Jupyter Notebook.

Stay in touch !

@DataCleanic

www.DataCleanic.ml