**All Machine Learning Algorithms You Should Know for 2023**

**Intuitive explanations of the most popular machine learning models**



First prize vector created by jcomp — [www.freepik.com](http://www.freepik.com)

If you enjoy this article and want to support me, [**Subscribe and become a member today**](https://terenceshin.medium.com/membership)to never miss another article on data science guides, tricks and tips, life lessons, and more!

For the past few years, I’ve compiled what I believed were the most important machine learning algorithms based on my experiences at work, my conversations with other data scientists, and what I read online.

This year, I want to expand on last year’s article by providing more **types** of models, as well as more models within each category. Through this, I hope to provide a repository of tools and techniques that you can bookmark so that you can tackle a variety of data science problems!

With that said, let’s dive into six of the most important types of machine learning algorithms:

1. Explanatory algorithms
2. Pattern mining algorithms
3. Ensemble learning algorithms
4. Clustering algorithms
5. Time series algorithms
6. Similarity algorithms

***If you want to test out these ML algorithms, check out*** [***Saturn Cloud***](https://saturncloud.io/?utm_source=Medium+&utm_medium=TDS&utm_campaign=Terence+Shin&utm_term=top-ML-algorithms-2023)***, a scalable, flexible data science platform with free and enterprise tiers.***

https://miro.medium.com/max/700/0*iu028PVERQ4h8mcZ.png

**1. Explanatory Algorithms**

One of the biggest problems in machine learning is understanding how various models get to their end predictions. We often know the “what” but struggle to explain the “why”.

Explanatory algorithms help us identify the variables that have a meaningful impact on the outcome we are interested in. These algorithms allow us to understand the relationships between the variables in the model, rather than just using the model to make predictions about the outcome.

There are several algorithms that you can use to better understand the relationships between the independent variables and the dependent variable for a given model.

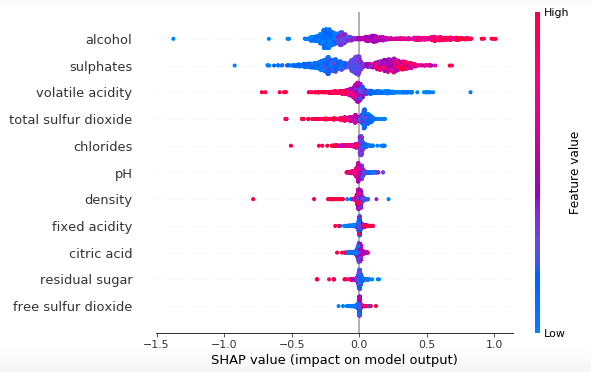


Image created by Author

**Algorithms**

* **Linear/Logistic Regression**: a statistical method for modeling the linear relationship between a dependent variable and one or more independent variables — can be used to understand the relationships between variables based on the t-tests and coefficients.
* **Decision Trees**: a type of machine learning algorithm that creates a tree-like model of decisions and their possible consequences. They are useful for understanding the relationships between variables by looking at the rules that split the branches.
* **Principal Component Analysis (PCA)**: a dimensionality reduction technique that projects the data onto a lower-dimensional space while retaining as much variance as possible. PCA can be used to simplify the data or to determine feature importance.
* **Local Interpretable Model-Agnostic Explanations (LIME)**: an algorithm that explains the predictions of any machine learning model by approximating the model locally around the prediction by constructing a simpler model using techniques such as linear regression or decision trees.
* **Shapley Additive explanations (SHAPLEY)**: an algorithm that explains the predictions of any machine learning model by computing the contribution of each feature to the prediction using a method based on the concept of “marginal contribution.”. It can be more accurate than SHAP in some cases.
* **Shapley Approximation (SHAP)**: a method for explaining the predictions of any machine learning model by estimating the importance of each feature in the prediction. SHAP uses a method called the “coalitional game” method to approximate Shapley values and is generally faster than SHAPLEY.

**2. Pattern Mining Algorithms**

Pattern mining algorithms are a type of data mining technique that are used to identify patterns and relationships within a dataset. These algorithms can be used for a variety of purposes, such as identifying customer buying patterns in a retail context, understanding common user behaviour sequences for a website/app, or finding relationships between different variables in a scientific study.

Pattern mining algorithms typically work by analyzing large datasets and looking for repeated patterns or associations between variables. Once these patterns have been identified, they can be used to make predictions about future trends or outcomes or to understand the underlying relationships within the data.

**Algorithms**

* **Apriori algorithm**: an algorithm for finding frequent item sets in a transactional database — it’s efficient and widely used for association rule mining tasks.
* **Recurrent Neural Network (RNN)**: a type of neural network that is designed to process sequential data as they are able to capture temporal dependencies in the data.
* **Long Short-Term Memory (LSTM)**: a type of recurrent neural network that is designed to remember information for longer periods of time. LSTMs are able to capture longer-term dependencies in the data and are often used for tasks such as language translation and language generation.
* **Sequential Pattern Discovery Using Equivalence Class (SPADE)**: a method for finding frequent patterns in sequential data by grouping together items that are equivalent in some sense. This method is able to handle large datasets and is relatively efficient, but may not work well with sparse data.
* **PrefixSpan**: an algorithm for finding frequent patterns in sequential data by constructing a prefix tree and pruning infrequent items. PrefixScan is able to handle large datasets and is relatively efficient, but may not work well with sparse data.

If you enjoy this article and want to support me, [**Subscribe and become a member today**](https://terenceshin.medium.com/membership)to never miss another article on data science guides, tricks and tips, life lessons, and more!

**3. Ensemble Learning**

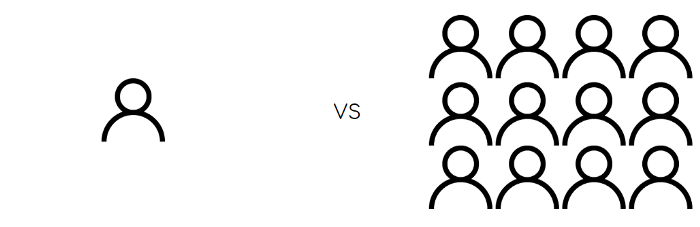


Image created by Author

Ensemble algorithms are machine learning techniques that combine the predictions of multiple models in order to make more accurate predictions than any of the individual models. There are several reasons why ensemble algorithms can outperform traditional machine learning algorithms:

1. Diversity: By combining the predictions of multiple models, ensemble algorithms can capture a wider range of patterns within the data.
2. Robustness: Ensemble algorithms are generally less sensitive to noise and outliers in the data, which can lead to more stable and reliable predictions.
3. Reducing overfitting: By averaging the predictions of multiple models, ensemble algorithms can reduce the tendency of individual models to overfit the training data, which can lead to improved generalization to new data.
4. Improved accuracy: Ensemble algorithms have been shown to consistently outperform traditional machine learning algorithms in a variety of contexts.

**Algorithms**

* [**Random Forest**](https://www.youtube.com/watch?v=J4Wdy0Wc_xQ): a machine learning algorithm that creates an ensemble of decision trees and makes predictions based on the majority vote of the trees.
* [**XGBoost**](https://www.youtube.com/watch?v=OtD8wVaFm6E&t=4s): a type of gradient boosting algorithm that uses decision trees as its base model and is known to be one of the strongest ML algorithms for predictions.
* [**LightGBM**](https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc): another type of gradient boosting algorithm that is designed to be faster and more efficient than other boosting algorithms.
* [**CatBoost**](https://towardsdatascience.com/why-you-should-learn-catboost-now-390fb3895f76): A type of gradient boosting algorithm that is specifically designed to handle categorical variables well.

**4. Clustering**

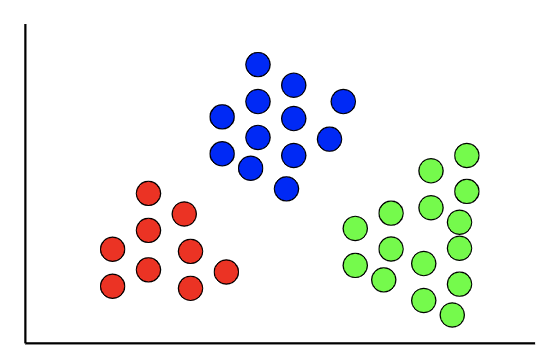


Image created by Author

Clustering algorithms are an unsupervised learning task and are used to group data into “clusters”. In contrast to supervised learning, where the target variable is known, there is no target variable in clustering.

This technique is useful for finding natural patterns and trends in data and is often used during the exploratory data analysis phase to gain further understanding of the data. Additionally, clustering can be used to divide a dataset into distinct segments based on various variables. A common application of this is in segmenting customers or users.

**Algorithms**

* **K-mode clustering**: a clustering algorithm that is specifically designed for categorical data. It is able to handle high-dimensional categorical data well and is relatively simple to implement.
* **DBSCAN**: A density-based clustering algorithm that is able to identify clusters of arbitrary shape. It is relatively robust to noise and can identify outliers in the data.
* **Spectral clustering**: A clustering algorithm that uses the eigenvectors of a similarity matrix to group data points into clusters. It is able to handle non-linearly separable data and is relatively efficient.

***If you want to test out these ML algorithms, check out*** [***Saturn Cloud***](https://saturncloud.io/?utm_source=Medium+&utm_medium=TDS&utm_campaign=Terence+Shin&utm_term=top-ML-algorithms-2023)***, a scalable, flexible data science platform with free and enterprise tiers.***

https://miro.medium.com/max/700/0*iu028PVERQ4h8mcZ.png

**5. Time Series Algorithms**

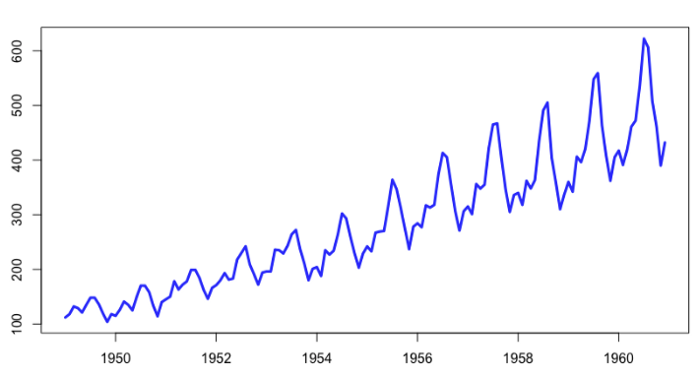


Image created by Author

Time series algorithms are techniques used to analyze time-dependent data. These algorithms take into account the temporal dependencies among the data points in a series, which is especially important when trying to make predictions about future values.

Time series algorithms are used in a variety of business applications, such as predicting demand for a product, forecasting sales, or analyzing customer behaviour over time. They can also be used to detect anomalies or changes in trends in the data.

**Algorithms**

* **Prophet time series modelling**: A time series forecasting algorithm developed by Facebook that is designed to be intuitive and easy to use. Some of its key strengths include handling missing data and trend changes, being robust to outliers, and being fast to fit.
* **Autoregressive Integrated Moving Average (ARIMA)**: A statistical method for forecasting time series data that models the correlation between the data and its lagged values. ARIMA can handle a wide range of time series data but can be more difficult to implement than some other methods.
* **Exponential smoothing**: a method for forecasting time series data that uses a weighted average of past data to make predictions. Exponential smoothing is relatively simple to implement and can be used with a wide range of data, but may not perform as well as more sophisticated methods.

**6. Similarity Algorithms**

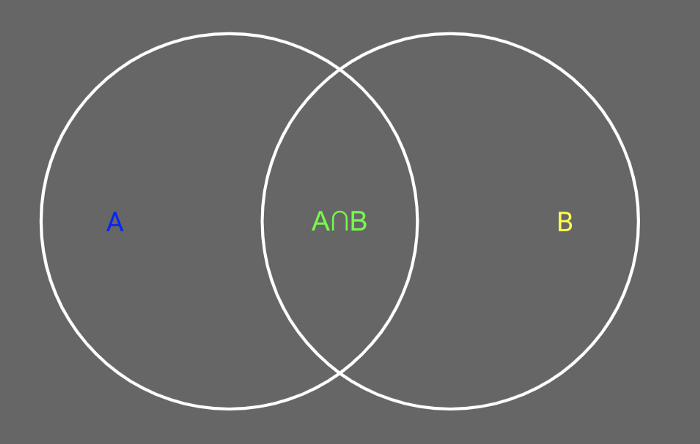


Image created by Author

Similarity algorithms are used to measure the similarity between pairs of records, nodes, data points, or text. These algorithms can be based on the distance between two data points (e.g. Euclidean distance) or on the similarity of text (e.g. Levenshtein Algorithm).

These algorithms have a wide range of applications, but are particularly useful in the context of recommendations. They can be used to identify similar items or suggest related content to users.

**Algorithms**

* [**Euclidean Distance**](https://www.youtube.com/watch?v=Dd16LVt5ct4&t=90s): a measure of the straight-line distance between two points in a Euclidean space. Euclidean distance is simple to calculate and is widely used in machine learning, but may not be the best choice in situations where the data is not uniformly distributed.
* [**Cosine Similarity**](https://www.youtube.com/watch?v=Dd16LVt5ct4): a measure of similarity between two vectors based on the angle between them.
* [**Levenshtein Algorithm**](https://medium.com/@appaloosastore/string-similarity-algorithms-compared-3f7b4d12f0ff): an algorithm for measuring the distance between two strings, based on the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other. The Levenshtein algorithm is commonly used for spell-checking and string matching tasks.
* [**Jaro-Winkler Algorithm**](https://medium.com/@appaloosastore/string-similarity-algorithms-compared-3f7b4d12f0ff): an algorithm for measuring the similarity between two strings, based on the number of matching characters and the number of transpositions. It’s similar to the Levenshtein algorithm and is often used for record linkage and entity resolution tasks.
* [**Singular Value Decomposition (SVD)**](https://www.youtube.com/watch?v=P5mlg91as1c&t=34s): a matrix decomposition method that decomposes a matrix into a product of three matrices — it’s an integral component for state-of-the-art recommendation systems.