Theory: Introduction to regression

18 minutes 0 / 5 problems solved Skip this topic Start practicing

58 users solved this topic. Latest completion was about 9 hours ago.

In this topic, you will learn more about regression problems. We will cover some real-world examples of regression and popular algorithms to solve them, as well as the most commonly-used evaluation metrics for such models.

§1. What is regression

In supervised machine learning, as well as in statistics, **regression** refers to predicting the value of a numerical, or continuous output Y from a number of predictor variables $X_1, X_2, ..., X_m$.

The term itself was coined in the nineteenth century by a British statistician Francis Galton, who was studying the dependency between the heights of descendants and that of their ancestors and noticed that descendants of tall ancestors tend to decrease, or ***regress***, down to the average human height.

Regression has a lot of real-world applications. Such problems are very common in finance. Think of, for instance, predicting exchange rates or market prices from those of couple of days ago, as well as the values of other indicators.

Another example of a regression problem is predicting flight ticket prices. If you ever booked a flight, you know that prices are formed dynamically by the airlines and change very often. Being able to predict the prices for the upcoming days is crucial for many booking services.

One more example of a regression problem is estimating the time it will take to deliver an order. For instance, each time you order online from your favorite restaurant, the system tells you when your order will be at your door.

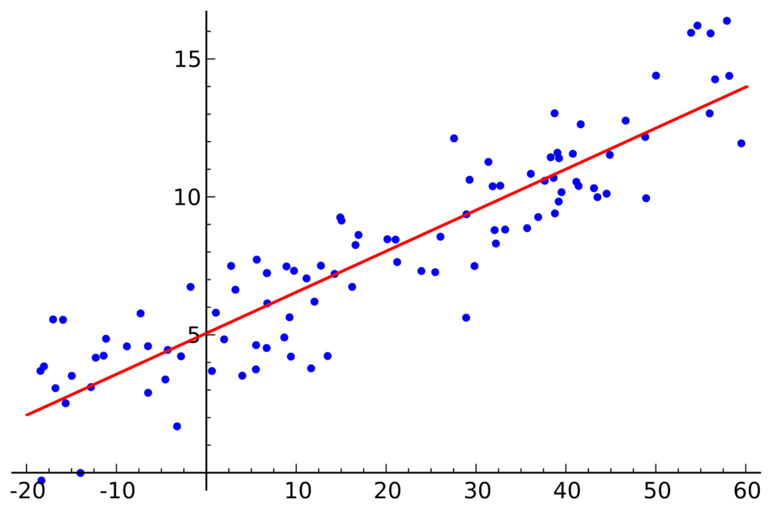
Alright, but how to solve all these problems? Here are some of the most-used techniques.

§2. Popular algorithms to solve regression problems

Arguably the simplest approach to solve regression problems is **linear regression**, which models the output Y as a linear combination of the inputs $X_1, X_2, ..., X_m$:

$$Y = \alpha_o + \alpha_1 \cdot X_1 + \alpha_2 \cdot X_2 + ... + \alpha_m \cdot X_m$$

Fitting a linear regression model refers to finding the optimal values of the model coefficients $\alpha_0, \alpha_1, ..., \alpha_m$ from the data. Below is an example of a one-dimensional case, where blue dots represent the data, and the red line corresponds to the predictions produced by the linear regression model (picture source):



Current topic:

Introduction to regression

Topic depends on:

✕ Typical ML pipeline

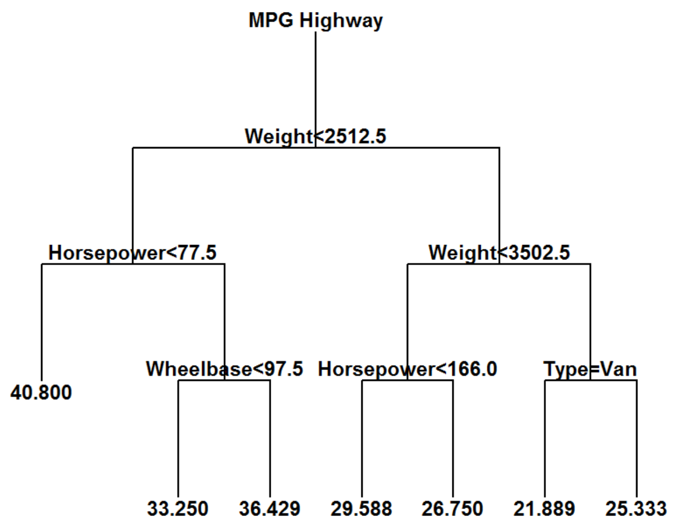
Topic is required for:

Decision Trees

Table of contents:

Unfortunately, not every problem can be solved with linear regression. In many cases, the dependency between the observed features and the output is non-linear. Therefore, more sophisticated techniques are needed to model it correctly.

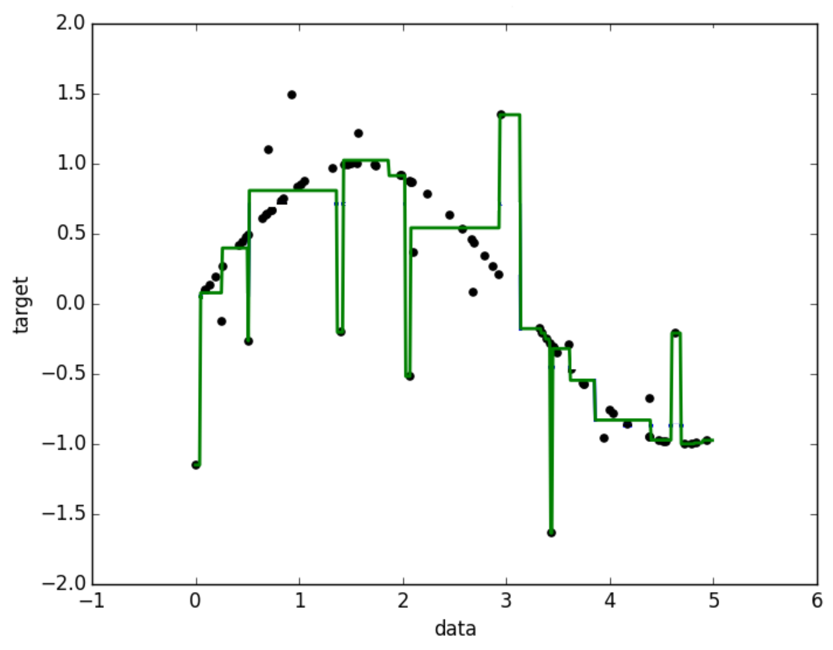
One model that is employed quite often is so-called **regression tree**. This model produces the prediction for the output variable by asking a series of questions about the values of the input features. Here is an example of a regression tree that predicts a vehicle's fuel consumption based on its characteristics (picture source):



Going from the top of the tree all the way to its leaf nodes by sequentially answering the questions like 'is weight less than 2512.5' and 'is type a van', we can obtain a prediction for each particular car. Building a regression tree model refers to deciding on which questions to ask.

What's the difference between a regression tree and a linear regression?

As you have just seen, predictions of a linear regression model lie on a straight line (or on a hyper-plane in a multi-dimensional case). Regression trees, by contrast, approximate the dependency between the input features and the target with a step function (picture source):



This already helps to capture a more complex relationship bewteen the inputs and the output. In practice, however, a single regression tree is rarely enough to get a good predictor for the output variable, as the model is still too simplistic. Luckily, there are ways to combine several regression trees in an **ensemble** that, in turn, produces fairly accurate joint predictions.

Both linear regression and single regression trees are *interpretable* models, meaning that it's easy for humans to explain where a prediction is coming from.

There are also other models that can be used to solve regression problems, for example **support vector regression** and **neural networks**. Such models are more expressive than linear regression or regression trees. At the same time, they are not interpretable and are often referred to as *black-box models* because of that.

§3. Evaluation metrics for regression algorithms

When solving a regression problem, you will most likely want to try out different algorithms and compare them with each other in order to determine the best one.

To do so, you will need to be able to evaluate the performance of a regression model.

In essence, you will need to assess how much the predictions $\hat{y}_1, \hat{y}_2, ..., \hat{y}_n$ produced by your model are different from the true values of the output variable $y_1, y_2, ...y_n$.

In machine learning, the most common way to do so is to compute so-called Mean Squared Error (MSE). MSE is the average squared deviation of the prediction from the true value, computed across all available examples:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The lower the MSE, the better the quality of the prediction.

Considering the squares of the difference between true and predicted values helps MSE give higher weight to larger errors. Indeed, small numbers get even smaller when squared (e.g. $0.01^2 = 0.0001$), so small prediction errors will have a limited impact on the MSE score. Large numbers, in turn, get much larger when squared (e.g. $100^2 = 10000$), so large errors will influence the MSE score a lot.

In fact, MSE is not only used to evaluate regression models but also to train them: optimal values of model parameters are often determined by minimizing MSE.

The downside of MSE is that it's difficult to interpret. Sometimes, a Root Mean Squared Error (RMSE) is used instead, which is nothing but a square root of MSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE behaves similar to MSE and can be interpreted as an average distance between the prediction and the rue point.

One disadvantage of RMSE is that, just like MSE, it's still scale-dependent. To facilitate the comparison between datasets or models with different scales, RMSE score is often normalized, for example, by the average value of the target. The resulting score is referred to as normalized RMSE (nRMSE):

$$nRMSE = \frac{RMSE}{\bar{y}},\ \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Another alternative is Mean Absolute Error (MAE), which is the average absolute deviation of model's predictions from the true values:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE is conceptually simpler and arguably easier to interpret for a human. Note however that, unlike (R)MSE, MAE doesn't penalize large errors more than the small ones. Thus, if large prediction errors are particularly undesirable, (R)MSE is a better choice.

Besides, mathematical properties of the absolute value function make it difficult to use MAE to find the optimal parameter values at the training phase.

§4. Conclusions

- Regression is a sub-field of supervised ML which refers to predicting numerical outputs.
- Simple regression problems can be solved by Linear Regression models, while more difficult ones are typically approached with ensembles of regression trees.
- The most typical evaluation metrics for regression models are MSE, RMSE and MAE.

Report a typo

6 users liked this theory. 0 didn't like it. What about you?

😍 🙂 😐 🙁 😡

Start practicing

Comments (0) Hints (0) Useful links (0) Show discussion