About Machine Learning

Two of the most common classes of machine learning models are supervised and unsupervised ML models. The key difference is that with supervised models, we have labels, or in other words, the correct answers to whatever it is that we want to learn to predict.

In unsupervised learning, the data does not have labels.

The graph on the left is an example of the sort of problem that an unsupervised model might try to solve. You own a used car lot. Heye, you want to look at the months that a car remains on the lot and its price, and then group or cluster cars, to see whether a car is likely to sell quickly. Critically, there is no ground truth here; the car lot does not have one group of cars that it is trying to sell and another group of cars it plans to hold. Consequently, unsupervised problems are all about discovery, about looking at the raw data, and seeing if it naturally falls into groups. At first look, it seems that there are two distinct clusters or groups that could be separated nicely with a line.

In this course though, we'll be focused on supervised machine learning problems, like this one. The critical difference is that with supervised learning, we have some notion of a "label," or one characteristic of each data point that we care about a lot.

Typically, this is something we know about in historical data, but we don’t know in real time. We know other things, which we call predictors, and we want to use those predictors to predict the thing we don’t know.

For example, let's say you are a purchaser for the used car lot. You have historical data on vehicle purchase prices for cars and trucks and how much was spent on repairs before the vehicle could be sold. Now you are at an auction. And you are looking at a vehicle that is for sale. You know the asking price and you know whether it is a car or a truck. But you don't know how much the repairs are going to be. In the historical data, the repair expense is a label. You create a model to predict the repair expense from the asking price. Then you try to predict the repairs, in real time, based on historical data and the values you know for a specific vehicle. This will give you an indicator of whether the asking price is a good deal for the vehicle or if they are asking too much because of the repair expenses. You will have an indicator how to bid at the auction.

https://pixabay.com/illustrations/chevrolet-limousine-usa-pkw-2178828/

Regression and classification are supervised ML model types

Price	Repairs	Type	Accident	Year	Odometer
43k	1,012	Truck	No	2012	53,232
6k	830	Car	No	2003	140,091
18k	2,704	Car	Yes	2017	6,508
3k	3,153	Truck	No	1999	162,000
35k	205	Car	No	2015	23,122

Model 1
Regression

Model 2
Classification

Google Cloud Training and Certification

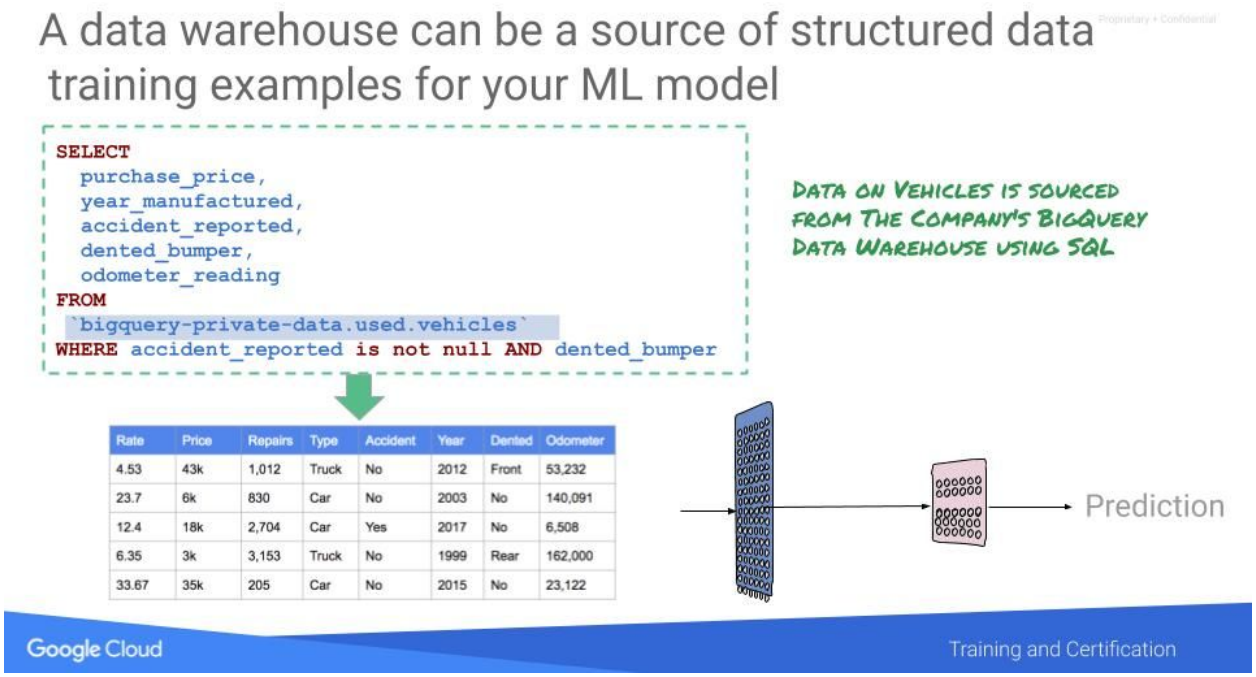
Within supervised ML there are two types of problems: regression and classification. To explain them, let's dive a little deeper into this data.

In this dataset of repair expenses, each row has many characteristics, such as type of vehicle, whether it has been involved in an accident, the year it was manufactured, and the odometer reading indicating how far it has traveled. We'll choose one of the columns as the characteristic we want to predict, called the "label". And we choose a set of the other columns which are called the "features".

In new model 1, we want to predict the repair amount. Therefore, the column "Repairs" is my label. I can use one, all, or any number of the other columns as my features to predict the repairs.
This will be a regression model because repairs is a continuous label.

In new model 2, we want to predict the whether the vehicle has been in an accident. Once again, I will use some set of the rest of the columns as features and try to predict the accident field. This will be a classification model because the label has a discrete number of values or classes.

https://pixabay.com/photos/pontiac-usa-wreck-dare-corrosion-2257344/



Now, where does this data come from? The tips dataset is what we call structured data -- consisting of rows and columns -- and a very common source of structured data for machine learning is your data warehouse. Unstructured data is things like pictures, audio, or video.

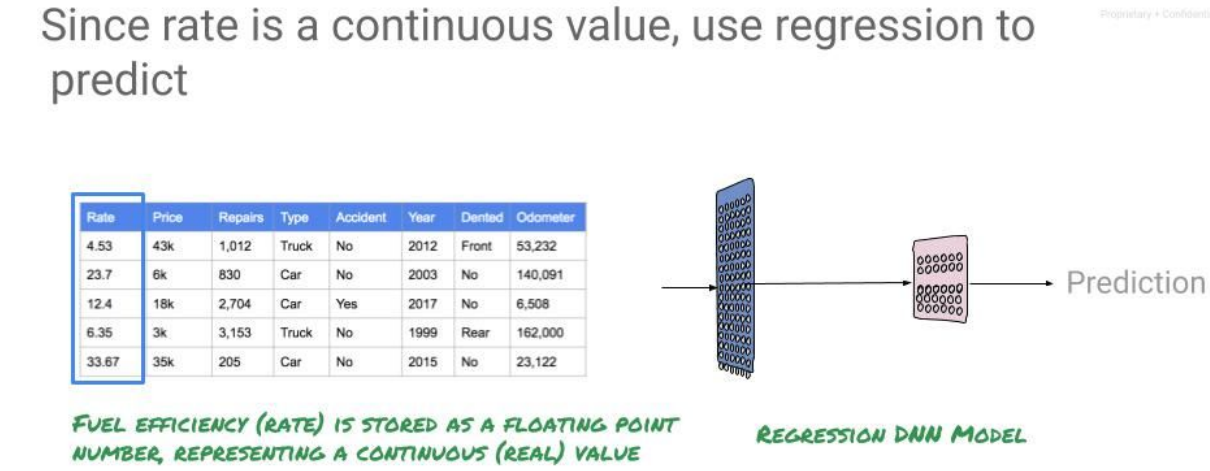
Here I'm showing you a vehicle dataset, a private dataset of company information. It is a dataset in BigQuery that is in your data warehouse

Let's say we want to predict the Odometer reading of the vehicle. In other words, we want to predict how far the vehicle has been driven.

You can do a SQL SELECT statement in BigQuery to create a ML dataset -- we will choose input features to the model, things like purchase price, accidents reported, manufacturing year, and type of vehicle.

Because the Odometer reading is a continuous value, this is a regression problem.

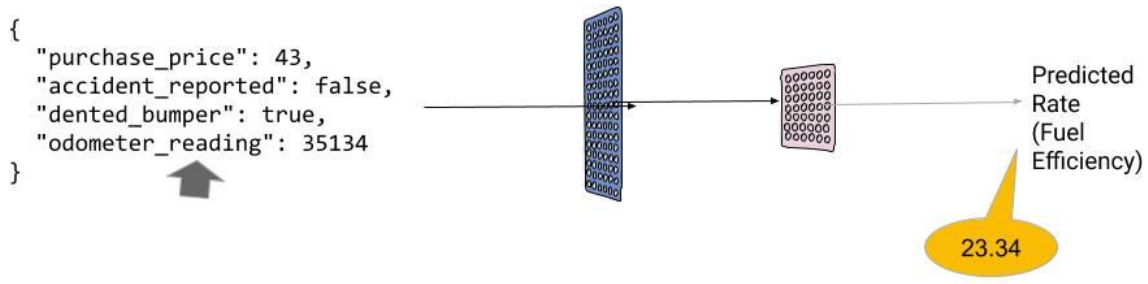
Making predictions from structured data is very commonplace, and that is what we focus on in the first part of the specialization.



Of course, this dataset can be used to predict other things too.

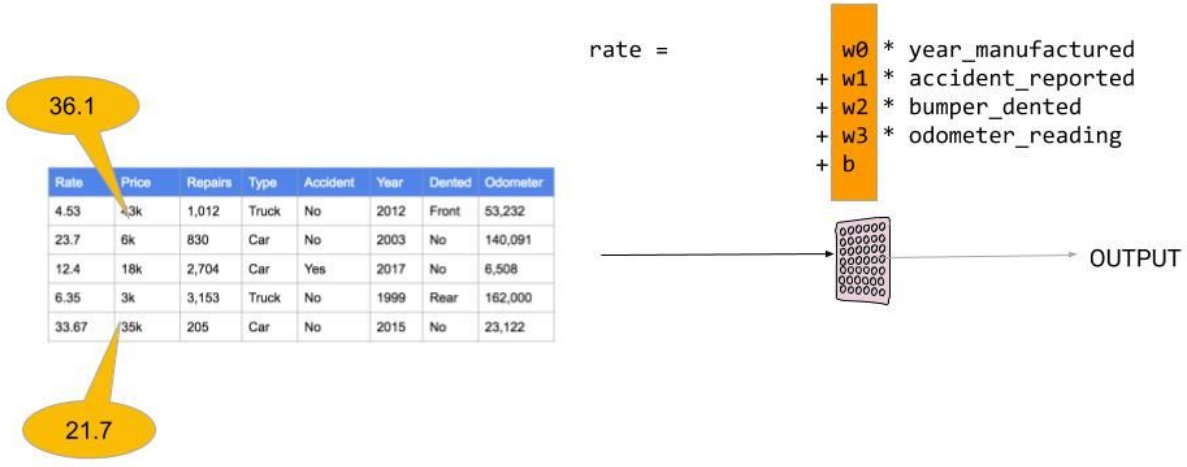
Perhaps we want to predict fuel efficiency (such as kilometers per liter), the rate at which fuel is consumed. And we want to predict this using other attributes of the data as features.
Fuel efficiency can be an indicator of the condition of the internal components like the engine and the drive train. So this could be a very useful value to predict.
The label here would be Rate, and it is a continuous variable. It is stored as a floating point number, which would make this a regression problem.

The model is fed information collected in real-time, and used for prediction



The key point here is that you are predicting from real-time data, for example, from Cloud Pub/Sub or your web application.
This information might originate from an online car auction service.

The model may even have only one layer



Training involves finding weights w0, w2, w2, w3 such that the predicted rate (fuel_efficiency) is really, really close to the labels.

A neural network with no hidden layers is a linear model …

Point out that the model here is a lot simpler than the one on the previous slides -- more layers in a DNN makes it more complex. Models on structured data are typically only a few layers deep whereas image classification DNNs can have hundreds of layers.