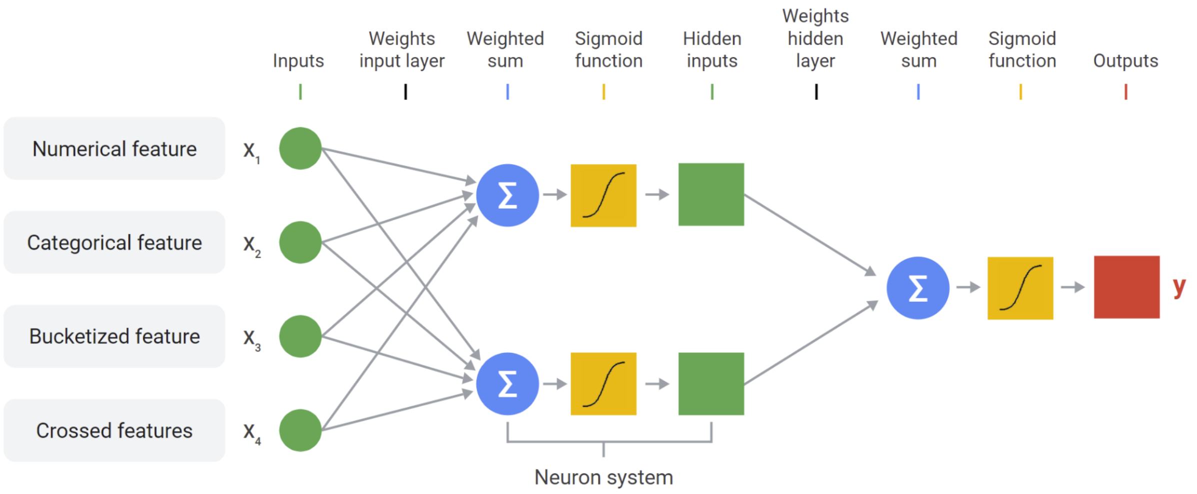
Defining “Drift” in Machine Learning Models

Gwendolyn Stripling, Ph.D., Michael Abel, Ph.D.

All opinions are our own.

Why do machine learning models lose their predictive power over time?



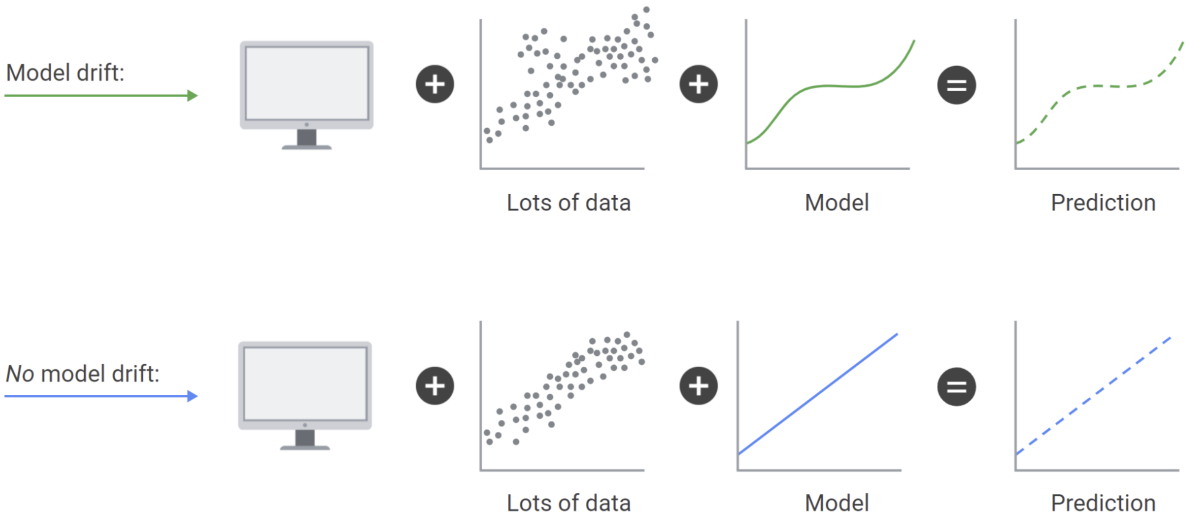
You’ll recall that machine learning models, such as neural networks, accept a feature vector and provide a prediction for our target variable *y* (as shown in the diagram above). These models learn in a supervised fashion where a set of feature vectors with expected output is provided.

Machine learning algorithms assumption

Remember that traditional machine learning algorithms were developed with certain assumptions.

1. Instances are generated at random according to some probability distribution *D*.
2. Instances are independent and identically distributed.
3. That *D* is stationary with fixed distributions.

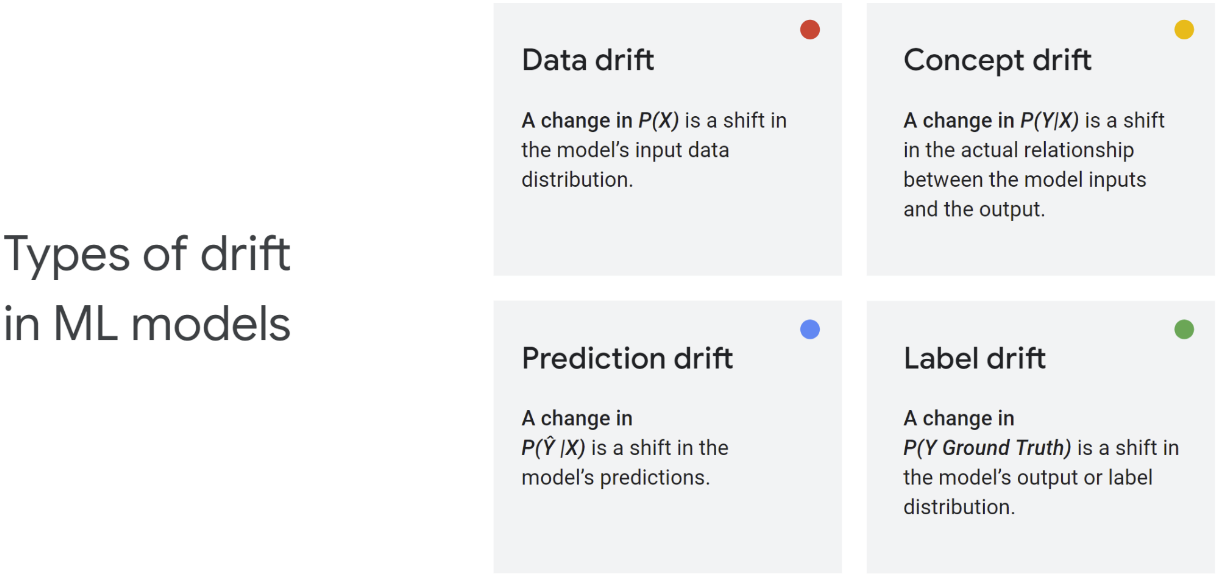
Drift is the change in an entity with respect to a baseline. In the case of production machine learning models, this is the change between the real-time production data and a baseline data set, likely the training set, that is representative of the task the model is intended to perform.



If your model were running in a static environment, using static or stationary data – for example data whose statistical properties do not change – then model drift wouldn't occur and your model would not lose any of its predictive power because the data you're predicting comes from the same distribution as the data used for training (as shown in the second example above). But production data can diverge or drift from the baseline data over time due to changes in the real world (as shown in the first example).

Types of drift in machine learning models

There are several types of drift in machine learning models.



Types of drift in ML models

Data drift

A change in *P(X)* is a shift in the model's input data distribution.

Concept drift

A change in *P(Y|X)* is a shift in the actual relationship between the model inputs and the output.

Prediction drift

A change in *P(Ŷ |X)* is a shift in the model's predictions.

Label drift

A change in *P(Y Ground Truth)* is a shift in the model's output or label distribution.

Data Drift or change in the distribution of X, $P(X)$, is a shift in the model's input data distribution. For example, incomes of all loan applicants decreases by 10% due to a health crisis -- such as a pandemic related to COVID-19. However, there is no change in how the loan is assessed.

Concept drift or change in the distribution of Y given X, $P(Y|X)$, is a shift in the actual relationship between the model inputs and the output. You get a sense of the interdependency between data drift and concept drift from the above example. If the incomes of all loan applicants decreases by 10% (due to macroeconomic factors impacted by the COVID-19 pandemic) then lending may be perceived as riskier, and thus a higher standard may be required to be eligible for a loan. In this case, an income level that was earlier considered creditworthy is no longer creditworthy, there is a change in how the loan is assessed.

Prediction drift or change in the predicted value of Y given X, $P(\hat{Y}|X)$, is a shift in the model's predictions. Now, given data drift or concept drift or the combination of the two, the number of credit-worthy applications for a loan product marketed in one zip code area has changed (e.g. car loans in an area where cars are a necessity but due to the pandemic, driving has decreased). Your model still holds.You can still predict the number of auto loans but you are unprepared for this scenario.

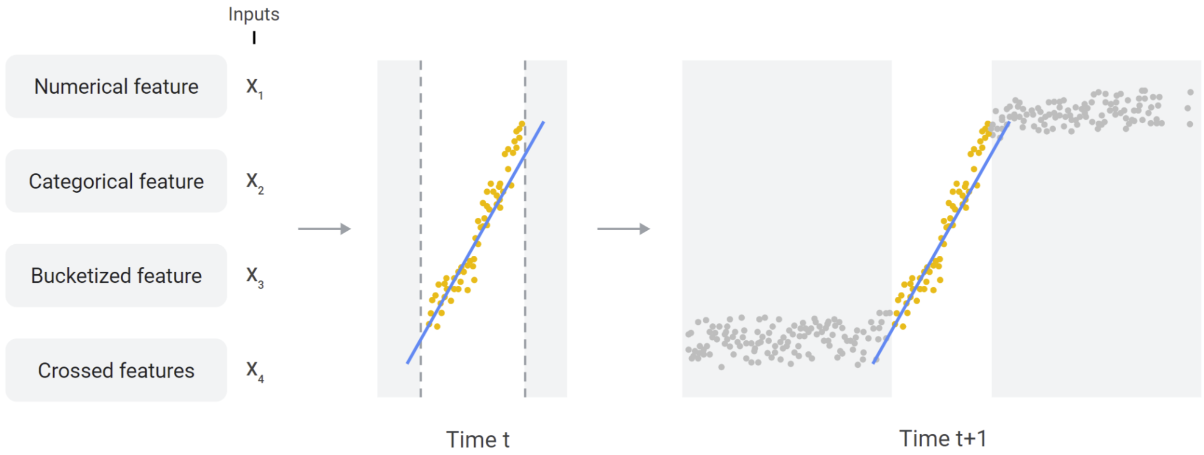
Label drift or change in the distrbution of Y as your target variable, or the distribution $P(Y)$, is a shift in the label distribution.

Data drift, feature drift, population, or covariate shift are all names to describe changes in the data distribution of the inputs. When data shift occurs, or when you observe that the model performs worse on unknown data regions, that means that the input data has changed.

Data drift
Feature drift
Population
Covariate shift
Describe changes in the data distribution of the inputs

The distribution of the variables is meaningfully different. As a result, the trained model is not relevant for this new data. It would still perform well on the data that is similar to the "old" one! The model is fine on the "old data", but in practical terms, it became dramatically less useful since we are now dealing with a new feature space. Indeed, the relationships between the model inputs and outputs have changed. This is illustrated in the example below.

Data drift

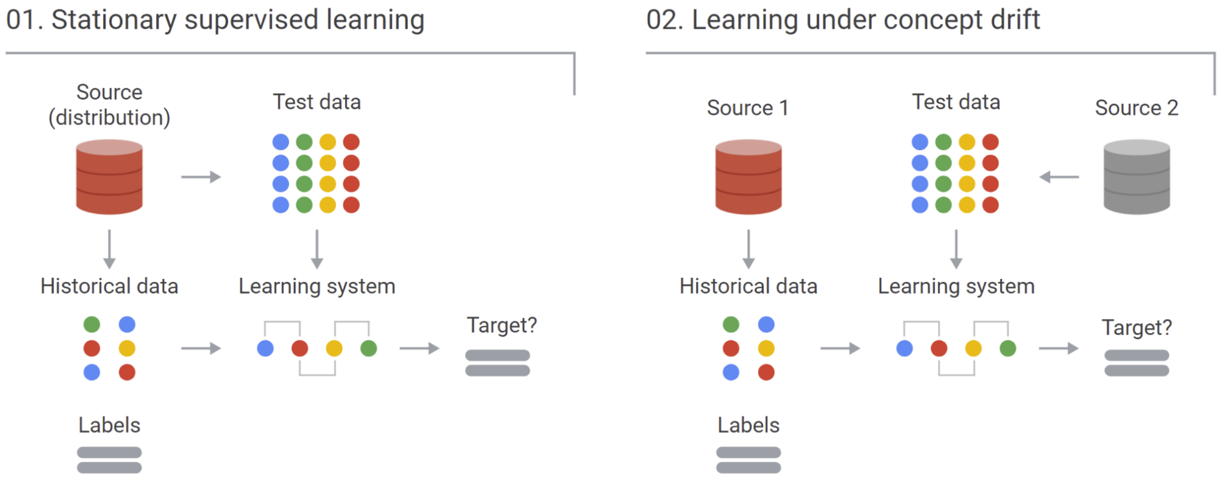


In contrast, concept drift occurs when there is a change in the relationship between the input feature and the label (or target).

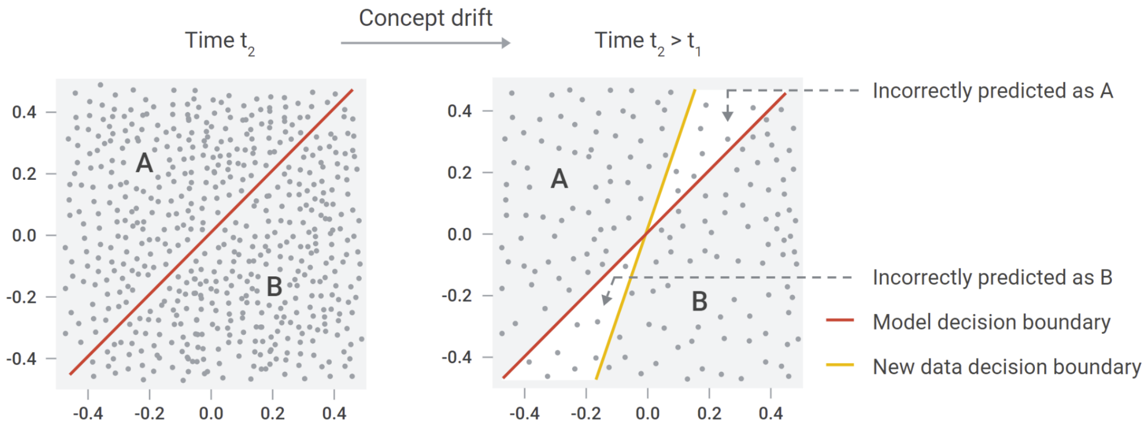
Let's explore an example of concept drift which highlights the change in the relationship between the input feature and the label. In this first example below, stationary supervised learning, historical data is used to make predictions. You might recall that in supervised learning, a model is trained from historical data and that data is used to make predictions.

This second example below is supervised learning under concept drift, where a new, secondary data source is ingested to provide both historical data and new data to make predictions. This new data could be in batch or real time. Whatever the form, it's important to know that the statistical properties of the target variable may change over time. As a result, an interpretation of the data changes with time, while the general distribution of the feature input may not.

This illustrates concept drift, where the statistical properties of the class variable (the target we want to predict) changes over time.



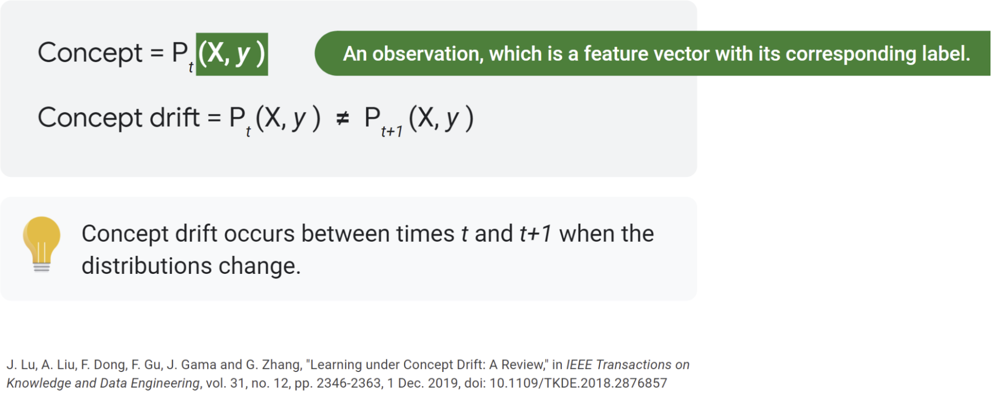
In this supervised learning classification example below, when the distribution of the label changes, it could mean that the relationship between features and labels is changing as well.



At the very least, it's likely that our model's predictions, which will typically match the distribution of the labels on the data on which it was trained, will be significantly less accurate.

Concept Drift - A probabilistic definition

Let's be a little more careful here with the definition of concept drift. Let's use X to denote a feature vector and y to denote its corresponding label. Of course, when doing supervised learning, our goal is to understand the relationship between X and y.



Concept = $P_t(X, y)$

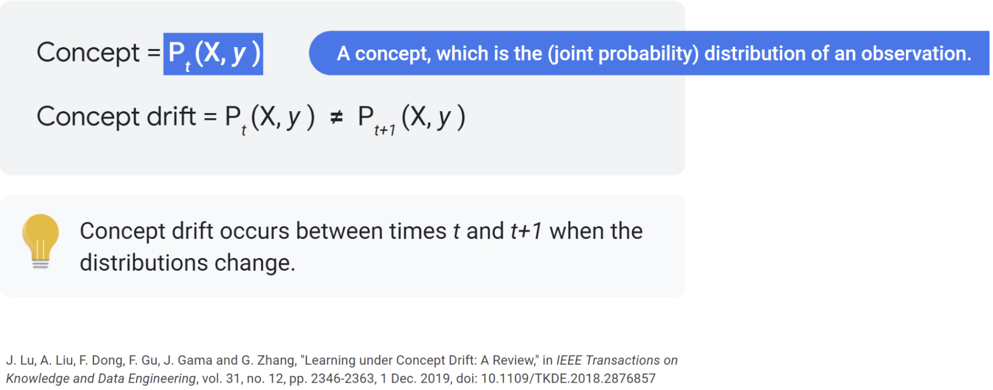
An observation, which is a feature vector with its corresponding label.

Concept drift = $P_t(X, y) \neq P_{t+1}(X, y)$

Concept drift occurs between times *t* and *t+1* when the distributions change.

J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857

We will define a concept as a description of the distribution of our observations. More precisely, you can think of this as a joint probability distribution of our observations. However, this concept could depend on time! Otherwise concept drift would be a non-issue, right?



Concept = $P_t(X, y)$

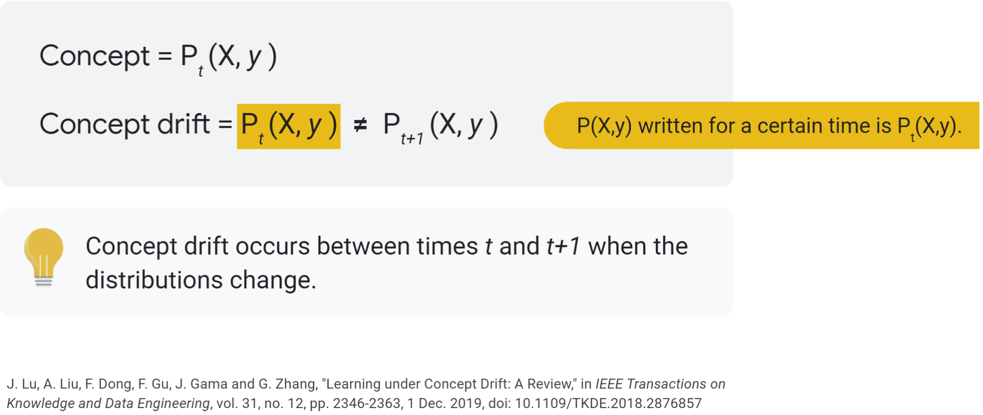
A concept, which is the (joint probability) distribution of an observation.

Concept drift = $P_t(X, y) \neq P_{t+1}(X, y)$

Concept drift occurs between times *t* and *t+1* when the distributions change.

J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857

We'll use the notation probability of X and y at time t P_t(X,y) when we want to consider the probability of X and y P(X,y) at a specific time.



Concept = $P_t(X, y)$

Concept drift = $P_t(X, y) \neq P_{t+1}(X, y)$

P(X,y) written for a certain time is P_t(X,y).

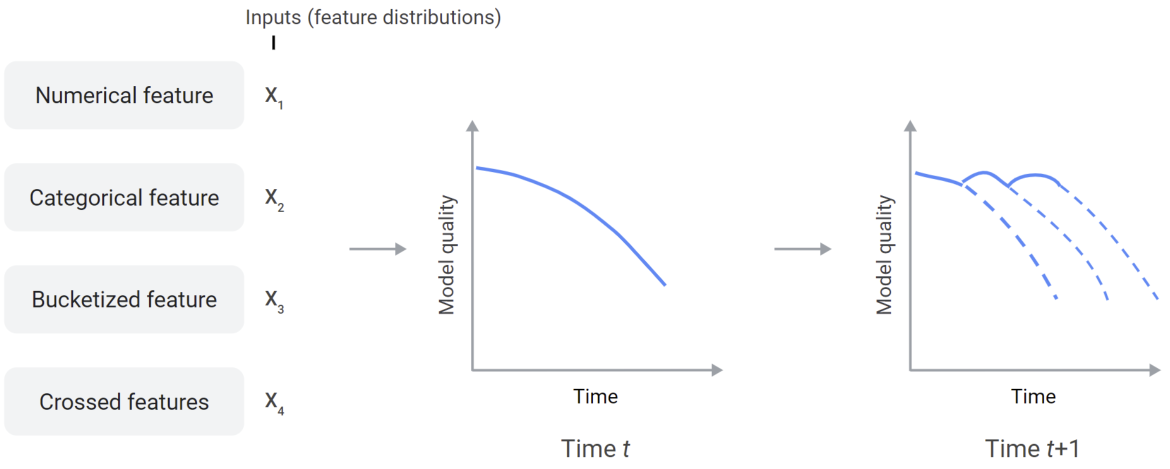
Concept drift occurs between times *t* and *t+1* when the distributions change.

J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857

Now it's easy to give a more rigorous description of concept drift. Simply put, concept drift occurs when the distribution of our observations shifts over time, or that the joint probability distribution we mentioned before changes.

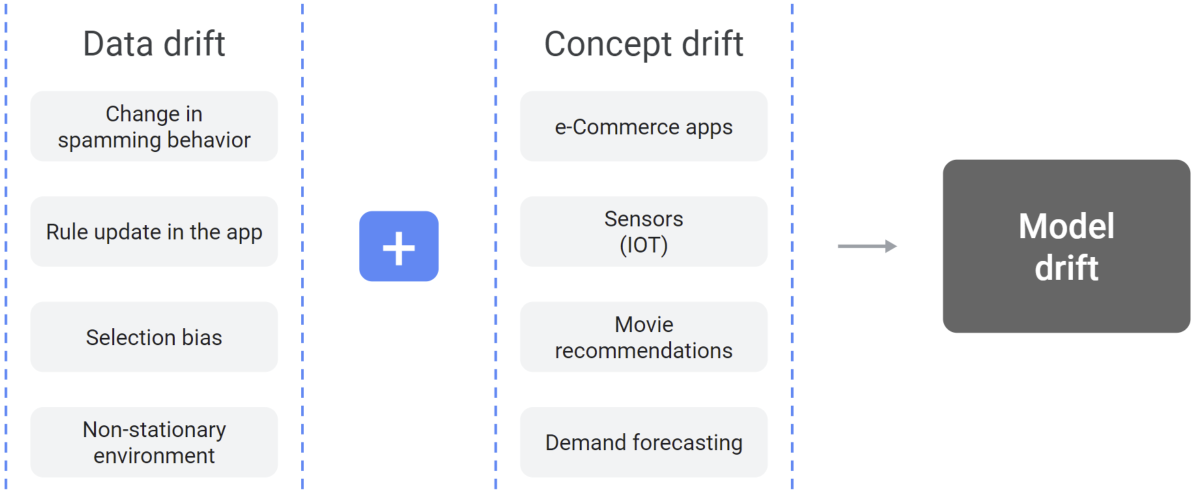
Remember, concept drift is the change in relationships between the model inputs and the model output.

Change in relationships between the model inputs and the model output



Examples of Data Drift and Concept Drift

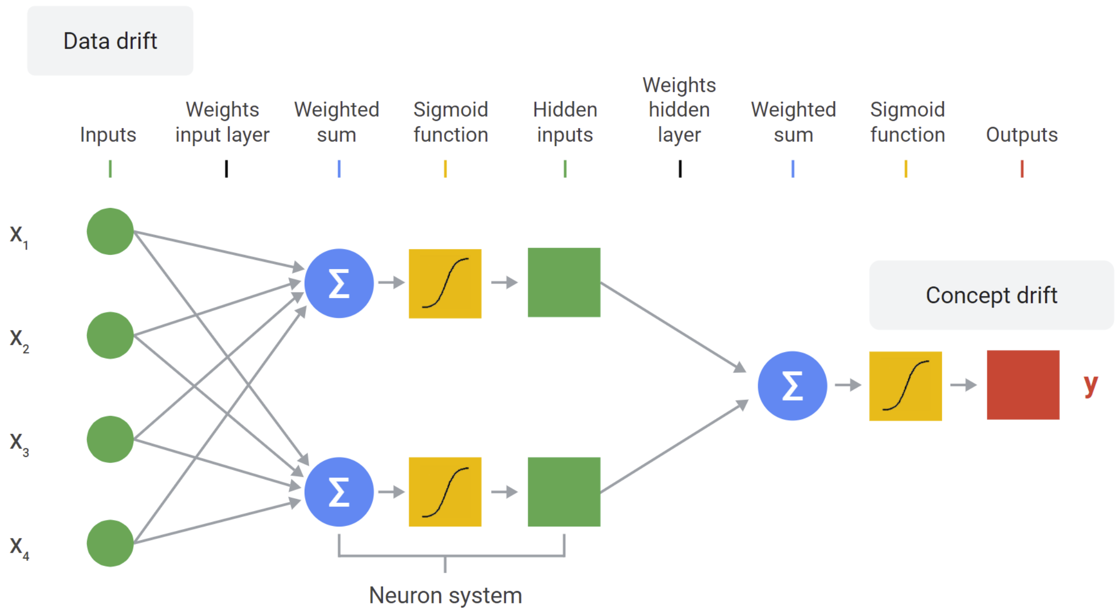
eCommerce apps are good examples of potential concept drift due to their reliance on personalization, for example, the fact that people's preferences ultimately do change over time. Sensors may also be subject to concept drift due to the nature of the data they collect and how it may change over time. Movie recommendations - again - similar to eCommerce apps - rely on user preferences - and they may change. Demand forecasting heavily relies on time, and as we have seen, time is a major contributor to potential concept drift.



As the world changes, your data may change. The change can be gradual, sudden, and seasonal. These changes will impact model performance.

Thus, machine learning models can be expected to degrade or decay. Sometimes, the performance drop is due to low data quality, broken data pipelines, or technical bugs.

Model drift = {Data drift, Concept drift}



Use TensorFlow Data Validation to check for drift and skew

For the majority of this blog, we’ve talked about data. TensorFlow Data Validation (TFDV) can be used to investigate and visualize your dataset. That includes looking at descriptive statistics, inferring a schema, checking for and fixing anomalies, and checking for drift and skew in our dataset. It's important to understand your dataset's characteristics, including how it might change over time in your production pipeline. It's also important to look for anomalies in your data, and to compare your training, evaluation, and serving datasets to make sure that they're consistent. Here is a lab on TensorFlow Data Validation in Google’s public github repository.

To learn more about this topic, please visit our Production Machine Learning Systems course on Coursera.