This module introduces a brief overview of supervised machine learning and its main applications: classification and regression. After introducing the concept of regression, you will learn its best practices, as well as how to measure error and select the regression model that best suits your data.

Learning Objectives:

* Identify the main types of Machine Learning
* Enumerate common day-to-day examples of Machine Learning applications
* Differentiate interpretation and prediction as the two main objectives of Machine Learning
* Identify statistical tests to assess the normality of outcome variables
* Identify best practices for Linear Regression, including how to measure error
* Become familiarized with the syntax to train a linear regression model and score a test set

## Course Prerequisites

**Basic Knowledge**

In order to be successful in this course, you will need a working knowledge of the following:

* Familiarity with programming on a Python development environment
* Familiarity with Jupyter notebooks
* Fundamental understanding of Calculus, Linear Algebra, Probability, and Statistics
* Familiarity with Exploratory Data Analysis, Feature Engineering, handling missing values, and handling categorical values.

Thinking humanly vs thinking rationally

A model is a small thing that captures a larger thing

A good model omits unimportant details while retaining what’s important

Ai in everyday life

Spam filtering, Web search, Postal mail routing, Fraud, movie recommendations, web adverts, social networks, speech recognition, vehicle driving assistance.

Types of ml

Supervised – data points have known outcome

Unsupervised – data points have kunknown outcome.

Semi-supervised – uses both data with outcomes and data without outcomes

ML Framework (applies to supervised machine learning models);;

Yp = ƒ(Ω,x)

X: input

Yp: value predicted by model

* Observations: rows or examples the model will see
* Features: the different ways we measure each observation. Th columns

Here we distinguish between yp (the prediction of our model) and y (the observed value of the target value.

X and y are arrays with 1 or more rows and columns. Y is a single column

Single observation can be represented by a row

Single feature can be represented by a column

Ω represents parameters of the model (1 or more variables)

Its important to sepcify exactly what yp, ƒ, Ω, and x are for any model we work with

Ω- represents parameters of the model ( 1 or more variables)

* This is what changes as the model learns
* Some models have many different parameters, some have very few.

As we implement our modeling approach, we will also select hyperparameters:

* A hyperparameter: is a parameter that is not learned directly from the data, but relates to implementation: training our ML model.
* We will apply techniques for using model performance to inform hyperparameter selection.

Our framework estimates a relationship between the features and target:

* Here Ω (the fit parameters) involve aspects of the model we estimate (fit) using the data
* To implement our approach, we make decisions regarding how to produce these estimates.
* These decisions lead to hyperparameters, that are an important part of the machine learning workflow (though not explicit components of the model).

Two main modeling approaches:

* Regression: y is numeric
  + e.g.: stock price, box office revenue, location(x,y coordinates).
* Classification: y is categorical
  + e.g.: face recognition, customer churn, which word comes next.

X: input

Yp: output (values predicted by the model).

Ƒ: prediction function that generates predictions from x and Ω

Data scientists train the model to find the best Ω given past experience.

J(y, yp): loss

* Most ml models define a quantitative score for how “good” our predictions are
* Typically measures how close our predictions are to the true values.

Update rule: using features x and outcome y, choose parameters Ω to minimize loss J.

Interpretation and Prediction

Interpretation:

* In some cases, the primary objective is to train a model to find insights from the data
* - in yp = ƒ(Ω,x), the interpretation approach uses Ω to give us insight into a system
* Common workflow:
  + Father , y; train model by finding the Ω that gives the best prediction yp = ƒ(Ω,x)
  + Focus on Ω (rather than yp) to generate insights
* Example interpretation exercises:
  + X = customer demographics, y = sales data; examine Ω to understand loyalty by segment.
  + X = car safety features, y = traffic accidents; examine Ω to understand what makes cars safer
  + X = marketing budget, y = movie revenue: examine Ω to understand marketing effectiveness

Prediction:

* In some cases, the primary objective is to make the best prediction.
* In yp = ƒ(Ω,x) the prediction approach compares yp with y
* The focus is on performance metrics, which measures the quality of the model’s predictions.
  + Performance metrics usually involves some measure of closeness between yp with y.
  + Without focusing on interpretability, we risk having a black-box model.
* Example prediction exercises:
  + X = customer purchase history, y = customer churn; focus on predicting customer churn
  + X = financial information, y = flagged default/non-default; focus on predicting loan default.
  + X = purchase history, y = next purchase; focus on predicting the next purchase.

Example: regression with housing data

In this course, we will examine housing datasets.

Here, our target is the price of housing and our features includes characteristics about the house and area

Suppose we fit out model y=ƒ(Ω,x) based on data on housing sales in Ames, Iowa, and obtain estimates in parameters Ω

These parameters represent coeffcients relating the features x with expected target values.

We can interpret our results to learn about feature importance.

Regression interpretation:

Examining results from regression of housing sales prices in ames, iowa.

Which features are most important:

* Overall quality
* Ground living area
* Year built

Here our target is the price of housing and our features include characteristics about the house and area.

Suppose we fit our model based on data on housing sales in ames, iow and obtain estimates of parameters Ω.

Our primary aim may be prediction, in which caese we are more focused on generating values for yp than in interpreting parameters.

Prediction using regression example:

Use characteristics to predict unknown sales prices, focusing on how accurately we are able to predict.

Customer churn occurs when a customer leaves a company.

Data related to churn may include a target variable for whether or not the customer left, as well as info on customer characteristics

Here we may be interested in both:

* Interpretation: understanding factors that may lead to customers leaving
* Prediction: estimating how long customers are likely to stay can help us understand how many we still need to support, and how valuable they are to the company

Two common approaches:

Interpretation and prediction in supervised ml

* Majority of projects will call for a balance.
* Interpretation can provide insights into improvements in prediction and vice-versa.
* Not all models will allow both: supervised ml models provide varying levels of support for interpretation vs prediction.

Machine learning is the subset of ai that focuses on model building to support a goal of interpretation and/or prediction.

Ml algorithms:

* Use past experience to build a model that is useful for future experience
* Follow a general form: yp = ƒ(Ω,x)

Regression vs. Classification Problems

Types of supervised learning

* Regression
  + Outcome is continuous (numerical)
* Classification
  + Outcome is a category

Supervised learning overview

Data with outcomes + model –> fit --> model

Data without outcomes + model --> predict --> predicted outcomes

Numeric predicting: movie revenue:

Movie data (known revenue + model --> fit --> model

Movie data (unknown revenue) + model --> predict --> predicted revenues

Classification: categorical answers:

Labeled data + model --> fit --> model

Unlabeled data + model --> predict--> predicted label

Classification: predicting spam emails

Emails labeled as spam (not spam) + model --> fit --> model

Unlabeled emails + model --> predict--> spam or not spam

What's needed for classification:

Model data with:

* Features that can be quantified.
* Labels that are known
* Method to measure similarity

Quiz:

Select the option that is th emost inaccurate regarding the difinition of machine learning:

* Ml is automated and requires no programming

This is the type of ml that uses both data with labeled outcomes and data without labeled outcomes

* Semi-supervised ml

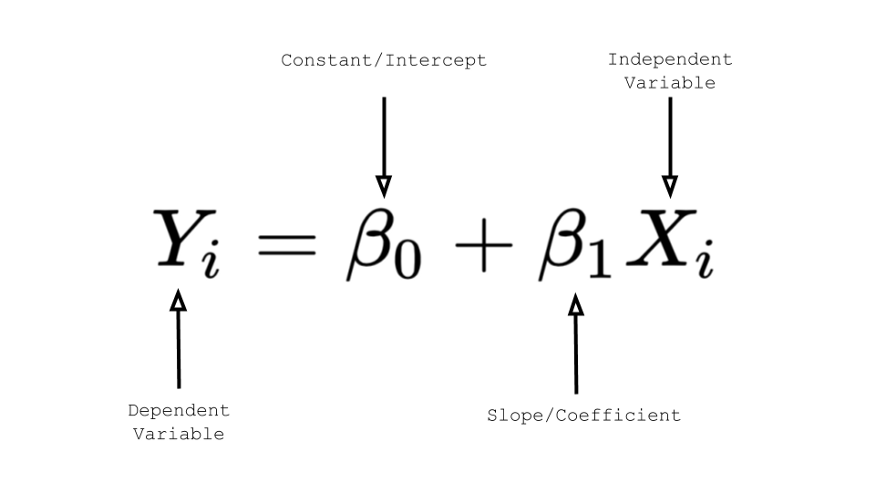
Predicting total revenue, number of customers, and percentage of returning customers are examples of:

* Regression

Predicting payment default, whether a transaction is fraudulent, and whether a customer will be part of the top 5% spenders on a given year, are examples of

* Classification

**Linear Regression**



Box office revenue = coefficient 0 + coefficient 1 \* movie budget

Box office revenue = 80million + 0.6 \* 160million

Use cost function to fit model

Develop multiple models

Compare results and choose best one

Sum of squared errors (sse):

Total sum of squares (tss)

Coefficient of determination(R^2): 1 - (sse/tss)

Import the class containing the regression method:

From sklearn.linear\_model import LinearRegression

Create an instance of the class

LR = LinearRegression()

Fit the instanc eon the data and then predict the expected value

LR = LR.fit(X\_train, y\_train)

Y\_predict = LR.predict(X\_test)

LINEAR REGRESSION DEMO

Value trying to predict is medv

Target is not normally distributed we can transform it.

Check for normal distribution.

* Historgram
* Statistical test

Higher the p value the closer the distribution is to normal. The lower the p value means we will reject the null hypothesis

Transform y variable to be normally distributed.

* Log transformation
* Square root transformation
* Box cox

Testing regression:

PolynomialFeatures ? (must set bias to false)

Train test split: hold out 30 percent to be the test state

Our train state will be 70 percent

The actual transformed values and the lambda value are the results from running the boxcox

Fit with linear regression

Predictions will be linear regression model

Because we transformed values with boxcox, we need to do an inverse transformation

Quiz:

Which statement about evaluating a ml model is the most accurate?

* Model estimation involves choosing parameters that minimize the cost function.

The unadjusted value from estimating a linear regression model will almost always increase if more features are added.

* True

The total sum of squares (tss) can be used to select the best-fitting regression model

* True

The sum of squared errors (sse) can be used to select the best-fitting regression model.

* True

## End of module review: Supervised Machine Learning and Linear Regression

### **Introduction to Supervised Machine Learning**

The types of supervised Machine Learning are:

* Regression, in which the target variable is continuous
* Classification, in which the target variable is categorical

To build a classification model you need:

* Features that can be quantified
* A labeled target or outcome variable
* Method to measure similarity

### 

### **Linear Regression**

A linear regression models the relationship between a continuous variable and one or more scaled variables. It is usually represented as a dependent function equal to the sum of a coefficient plus scaling factors times the independent variables.

Residuals are defined as the difference between an actual value and a predicted value.   
A modeling best practice for linear regression is:

* Use cost function to fit the linear regression model
* Develop multiple models

**Compare the results and choose the one that fits your data and whether you are using your model for prediction or interpretation.**   
  
**Three common measures of error for linear regressions are:**

* Sum of squared Error (SSE)
* Total Sum of Squares (TSS)
* Coefficient of Determination (R2)

### 

### **Linear Regression Syntax**

The most simple syntax to train a linear regression using scikit learn is:

* from sklearn.linear\_model import LinearRegression
* LR = LinearRegression()
* LR = LR.fit(X\_train, y\_train)

To score a data frame X\_test you would use this syntax:

* y\_predict = LR.predict(X\_test)

Final quiz

You can use supervised machine learning for all of the following examples, except:

* Segment customers by their demographics

The autocorrect on your phone is an example of:

* Supervised learning

This is the type of ml that uses both data with labeled outcomes and datat without labeled outcomes:

* Semi-supervised ml

This option describes a way of turning a regression problem into a classification problem:

* Create a new variable that flags 1 for above a certain value and 0 otherwise.

This is the syntax you need to preidct new data after you have trained a linear regression called LR:

* LR.predict(X\_test)

All of these optioins are useful error measures to compare regression:

* ROC index

It is less concerning to treate a ml model as a black box for prediction purpose compared to interpretation purposes.

* True.