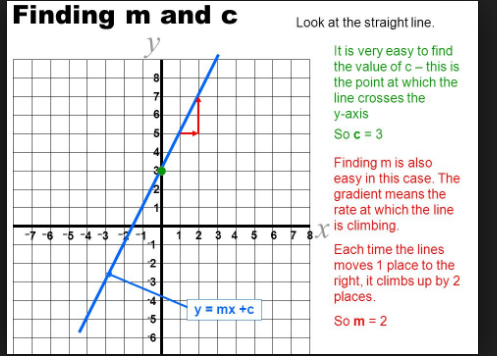
Linear regression:

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves

Linear:



Y=mx+c

M(slope) is tan(theta) =dy/dx

Slope is coeffictient, correlation( it shows the relation or proportionality between x and y)

C(intercept)-it is also called bias(error). The relation is biases when c!=0

Note: to determine m and c value two data sets are enough. Remaining data sets are used to remove the bias(error)

We have to identify the line which is a best fit with a minimal residual of the dataset.

y predicted and y real

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression. we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and nature of regression line is linear.

Y=a+b\*X + e, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable(s)

coefficients are estimated using the **least squares criterion**, which means we are find the line (mathematically) which minimizes the **sum of squared residuals** (or "sum of squared errors"):

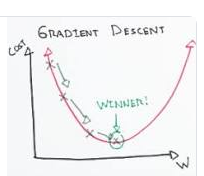
Gradient decent: adjust the coordinate of error with repect to m orc. rate of change with m or c.

De/dm, dc/dm

Z=error if loss=y-yhatt is not error

Local miniumn:adjust the error to get a point at which the error is zero.

**Gradient descent** is an optimization algorithm used to minimize some function by iteratively moving in the direction of **steepest descent** as defined by the negative of the **gradient**. In machine learning, we use **gradient descent** to update the parameters of our model.



Reinforcement learning:

Learning will be impacted with market dimensions in a win or loss move like in shares.probability to learn.learning from win or loss is reinforcement. Try to learn winning move.pr2 and Sophia built on these models.emit hyposonic signal will be generated.

Workflow:

1. Data preparation: selecting the data(features),cleaning data(nulls handling,normalizing(scaling the data), handaling categorical data)
2. Training set generation
3. Algorithm training –shift or decrease the loss
4. Development and monitioring

Accuracy:

Linear regression is high bias and low varience

A closely related concept is **confidence intervals**. Statsmodels calculates 95% confidence intervals for our model coefficients, which are interpreted as follows: If the population from which this sample was drawn was **sampled 100 times**, approximately **95 of those confidence intervals** would contain the "true" coefficient.

SSTSST = Sum of Squares Total, SSESSE = Sum of Squared Errors, and SSRSSR = Regression Sum of Squares. The equation in the title is often written as:

∑i=1n(yi−y¯)2=∑i=1n(yi−y^i)2+∑i=1n(y^i−y¯)2

SSE=

Perpendicular distance from y predicted and y.

Sum of square of these values is sum of square of errors

SSR: distance between y and y\_mean. Sum of square of these values is sum of square of rediduces, regression sum of squars.

R\*\*2= 1-SSR/SST

 R-squared is the **proportion of variance explained**, meaning the proportion of variance in the observed data that is explained by the model, or the reduction in error over the **null model**. (The null model just predicts the mean of the observed response, and thus it has an intercept and no slope.) R-squared is between 0 and 1

Statistical model:

import statsmodels.formula.api as smf

lm = smf.ols(formula='Sales ~ TV', data=data).fit()

lm.conf\_int()

two types:

backword elimination: taking full features sets and removing one by one and checking the R\*\*2

forward elimination: taking single feature and adding features one by one and checking the R\*\*2

OLS:ordinary least square method.

In this process we had a column to represent C(intercept).

Endog is lable and exog is features.

R\*\*2 and Adjusted R\*\*2 would determine accuracy. R\*\*2 value would not change with change in features but adjusted R\*\*2 determines the accuracy.

Adjusted R2= (1-R)\*\*2\*(n-1)/(n-p-1) where n is number of rows and p is number of columns

Evaluation matric would consider adjusted R\*\*2 as it depends on columns and rows.

How do I decide **which features to include** in a linear model? Here's one idea:

* Try different models, and only keep predictors in the model if they have small p-values.
* Check whether the R-squared value goes up when you add new predictors.

Multi linear regression:

If the features are continuous then no issues but if they are categorical then follow below:

1. Feature is binomial –means it has two values. Then it can be converted to numerical using labelencoder. For scikit-learn, we need to represent all data **numerically**. If the feature only has two categories, we can simply create a **dummy variable** that represents the categories as a binary value:
2. Features has more than two categories- first calculate the labelencoder and then to onehotencoder. We have to represent Area numerically, but we can't simply code it as 0=rural, 1=suburban, 2=urban because that would imply an **ordered relationship**between suburban and urban((and thus urban is somehow "twice" the suburban category).).i nstead, we create **another dummy variable**: Why do we only need **two dummy variables, not three?** Because two dummies captures all of the information about the Area feature, and implicitly defines rural as the baseline level. (In general, if you have a categorical feature with k levels, you create k-1 dummy variables.)

Lable encoder—converts to 0,1,2,3

Onehotencodeer: convers to 0 and 1’s converts tobits

Test size=25% default

Random state: fix the state of model while distribution oor division of data. Division of data based on the some parameter or patern.

Predict based on file not on method(object or model)

Use pickle or joblib: model is a mathematical equation and need to be saved in pickle or joblob.

<https://machinelearningmastery.com/linear-regression-for-machine-learning/>

<https://machinelearningmastery.com/category/statistical-methods/>

<https://machinelearningmastery.com/category/linear-algebra/>

<https://towardsdatascience.com/tagged/linear-regression>