Linear Regression

DATA FOLKZ
CATAPULT DATA LEADERS

Overview

- Linear Regression Model
- Measuring error
- Error Surface
- Regression Algorithm
- Example

Prerequisite

- Derivative
- Finding Derivatives
- Chain Rule
- Partial Differentiation
- Dataset, Descriptive Features, Target

Supervised Learning

- In a supervised learning problem, you have access to input variables (X) and outputs (Y), and the goal is to predict an output given an input.

- Examples:
  - Housing prices (Regression): predict the price of a house based on features (size, location, age etc)
  - Cat vs. Dog (Classification): predict whether a picture is of a cat or a dog

Regression

- Predicting a continuous outcome variable:
  - Predicting a company's future stock price using its profit and other financial information
  - Retail - How much will be the daily, monthly, and yearly sales for a given store for the next three years?
  - How much will be the monthly electricity cost for the next three years?
  - How many customers will claim the insurance this year?
  - What will be the temperature for the next five days?
  - Predicting annual rainfall based on local flora and fauna

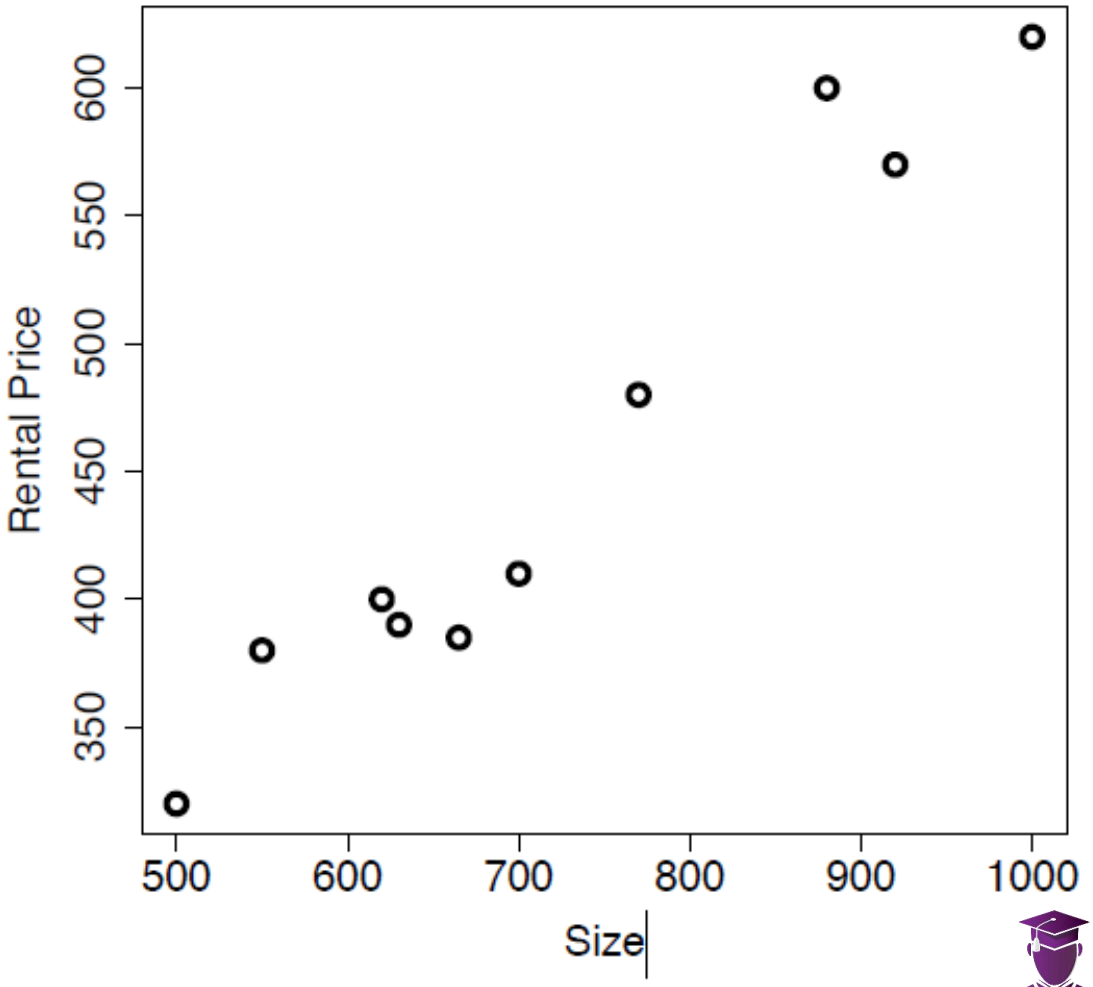
Ex: Predicting the rental price of an Office

A dataset that includes office rental prices and a number of descriptive features

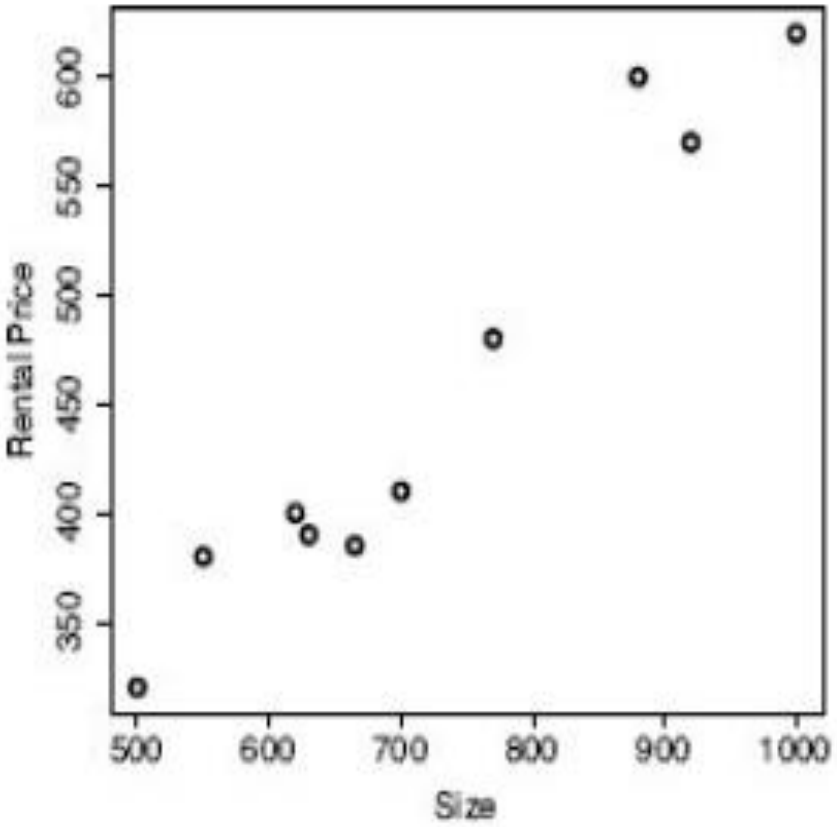
ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

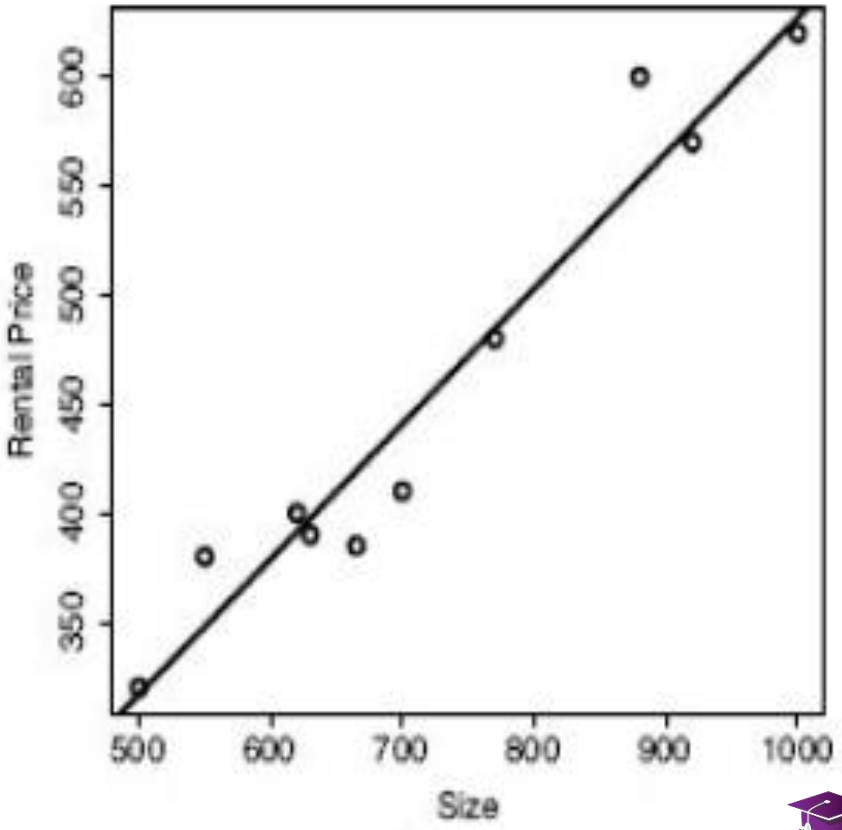
Scatter plot of the SIZE and RENTAL PRICE features

ID	SIZE	RENTAL PRICE
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620



Linear Model relating Size and Rent

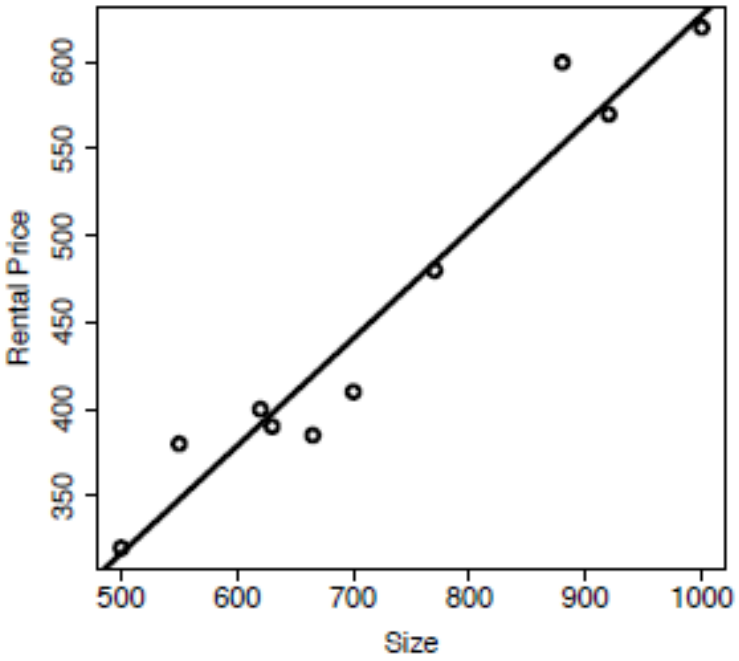




Linear Model

- The equation of a line can be written as: $y = mx + b$
- This model is:

$$\text{RENTAL PRICE} = 6.47 + 0.62 \ \text{SIZE}$$



Advantage

- Understand how office size affects office rental price.
- Determine the expected rental price for office sizes that we have never actually seen in the historical data
- How much would we expect for size = 730
.

Prediction

$$\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$$

- Using this model determine the expected rental price of the 730 square foot office:

$$\begin{aligned} \text{RENTAL PRICE} &= 6.47 + 0.62 \times 730 \\ &= 459.07 \end{aligned}$$

Regression Model

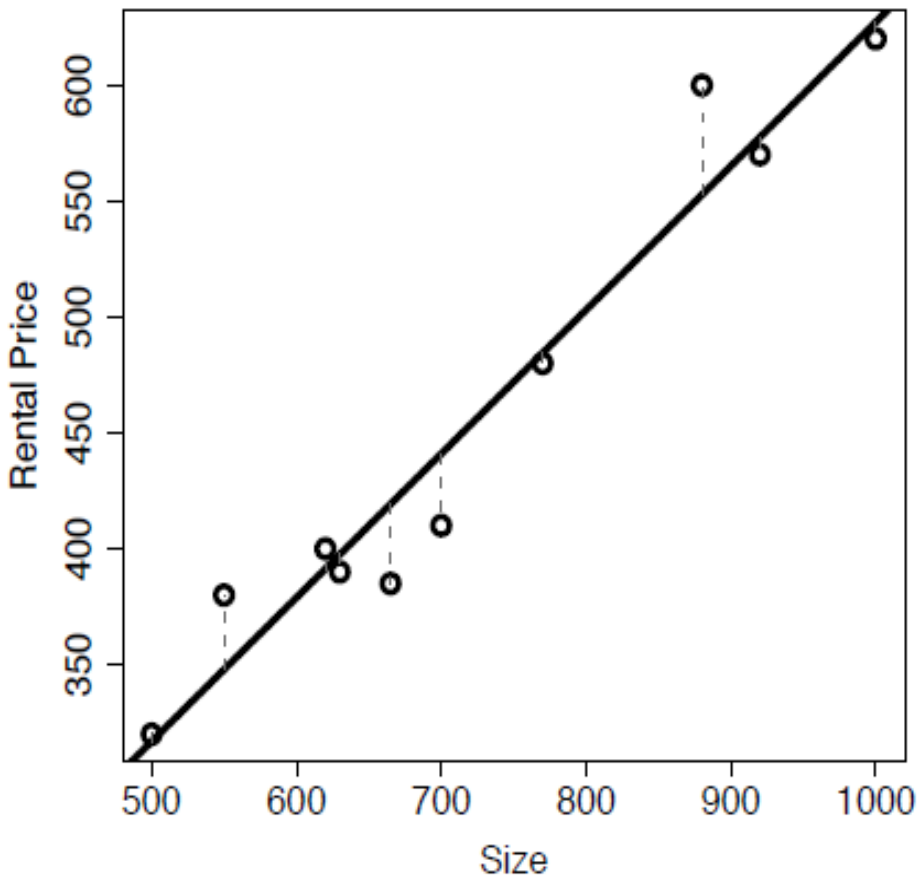
$$y = mx + b$$

$$y = w[0] + w[1] * size$$

Objective is :

- to determine the optimal values for the weights in the model
- To measure the error between the predictions a model makes and the actual rental prices

Measuring Error



$$Error\ \ E = \sum_{i=1}^{n} (t_i - y_i)^2$$

$$= \sum_{i=1}^{n} (t_i - (w[0] + w[1] * size_i))^2$$

.....

DATA FOLKZ
CATAPULT DATA LEADERS

Measuring Error

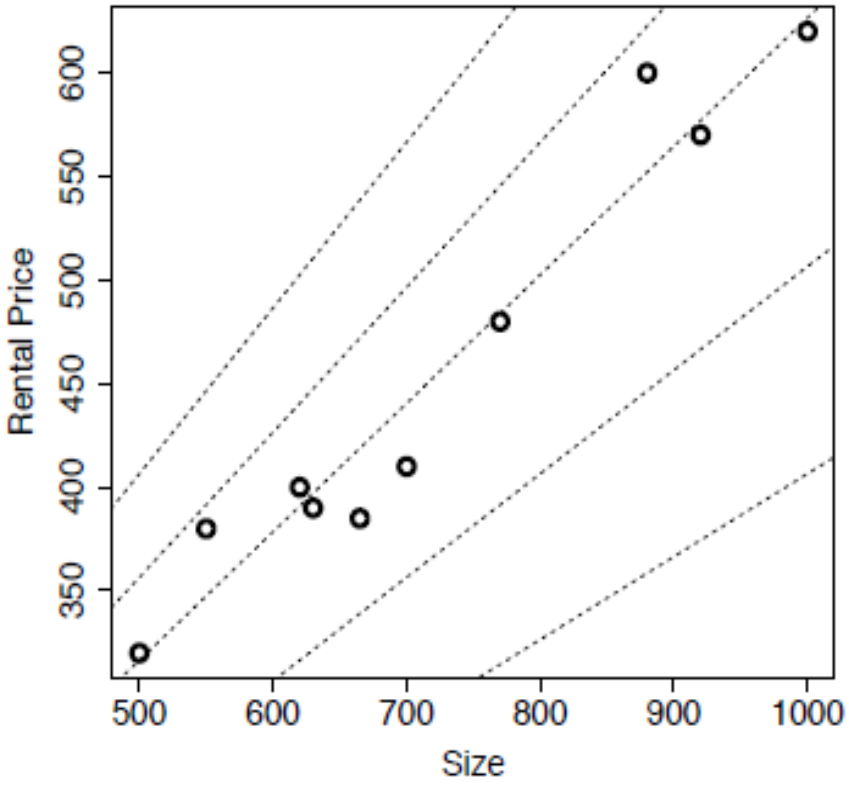
The sum of squared errors for the model (with **w[0] = 6.47 and w[1] = 0.62)**

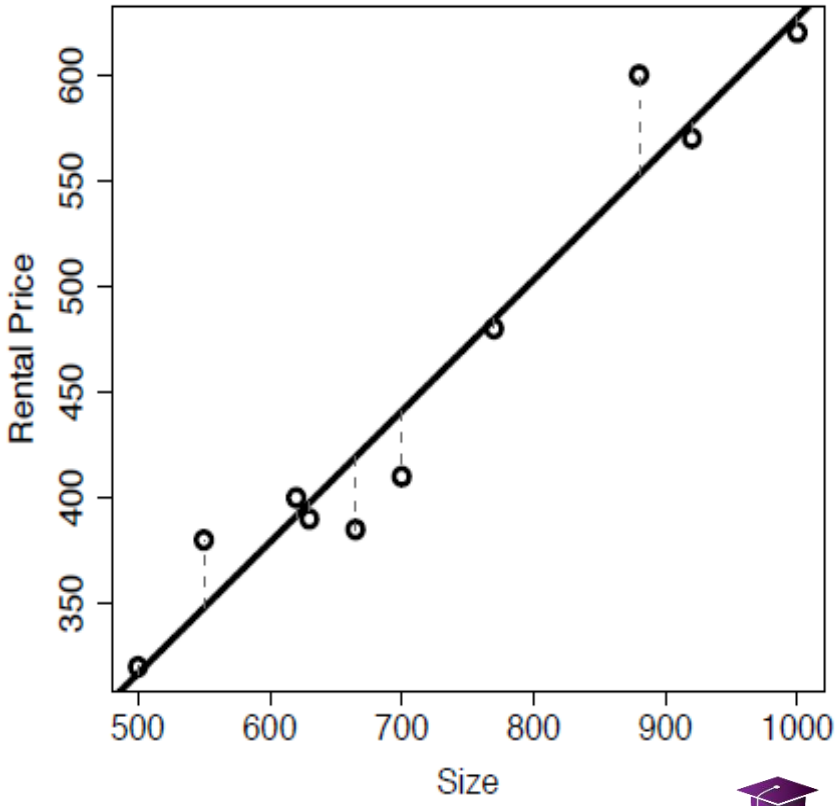
ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

ID	Rental Price	Model Prediction	Error Error	Squared Error
1	320	316.79	3.21	10.32
2	380	347.82	32.18	1,035.62
3	400	391.26	8.74	76.32
4	390	397.47	-7.47	55.80
5	385	419.19	-34.19	1,169.13
6	410	440.91	-30.91	955.73
7	480	484.36	-4.36	19.01
8	600	552.63	47.37	2,243.90
9	570	577.46	-7.46	55.59
10	620	627.11	-7.11	50.51
			Sum	**5,671.64**

Collection of simple linear models

With **w[1] set to 0.4, 0.5, 0.7, and 0.8, the sums of squared errors** are 136,218, 42,712, 20,092, and 90,978





Important Observation

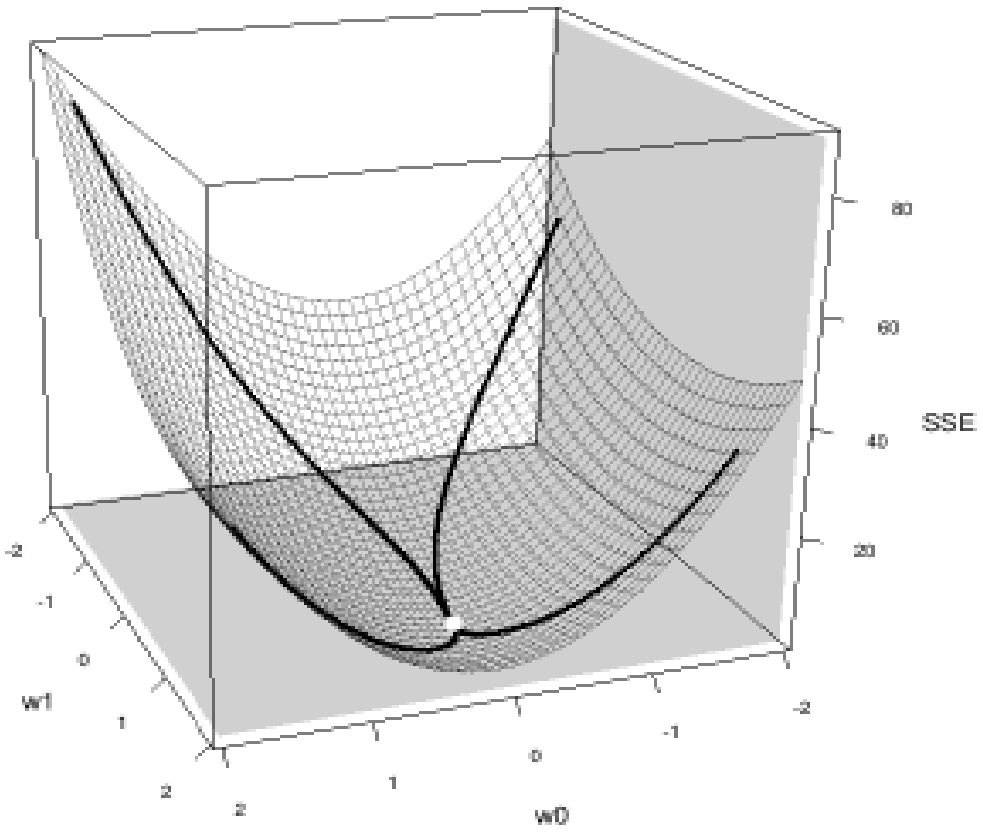
- The sum of squared errors function can be used to measure how well any combination of weights fits the instances in a training dataset

- From a collection of simple linear models we need to identify the one which minimizes the sum of the squared errors

- How do we identify this?

Error Surface

- For every possible combination of weights, **w[0] and w[1],** there is a corresponding sum of squared errors value that can be joined together to make a surface.



Least squares optimization.

- Having a global minimum means that on an error surface, there is a unique set of optimal weights with the lowest sum of squared errors.

- If we can find the global minimum of the error surface, we can find the set of weights defining the model that best fits the training dataset.

- This approach to finding weights is known as **least squares optimization.**

Stationary Points

- We can find the optimal weights at the point where the **partial derivatives of the error** surface with respect to **w[0] and w[1] are equal to 0**

$$E = \sum_{i=1}^{n} (t_i - (w[0] + w[1] * size_i))^2$$

$$\frac{\partial E}{\partial w[0]} = \sum_{i=1}^{n} 2(t_i - (w[0] + w[1] * size_i))(-1) \quad = 0$$

$$\sum_{i}^{n} t_i = n.w[0] + w[1] * \sum_{i}^{n} size_i$$

$$\frac{\partial E}{\partial w[1]} = \sum_{i=1}^{n} 2(t_i - (w[0] + w[1] * size_i))(size_i) \quad = 0$$

$$\sum_{i}^{n} t_i size_i = w[0] \sum_{i}^{n} size_i + w[1] * \left(\sum_{i}^{n} size_i \right)^2$$

Simple example

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

$$\sum_i^n t_i = n.w[0] + w[1] * \sum_i^n size_i$$

$$\sum_i^n t_i\, size_i = w[0] \sum_i^n size_i + w[1] * \left(\sum_i^n size_i\right)^2$$

$$4555 = 10w[0] + w[1] * 7235$$

$$3447725 = w[0] * 7235 + w[1] * 5479725$$

Simple example

- import numpy as np
- mat=np.array([[10, 7235], [7235, 5479725]])
- b=np.array([4555, 3447725])
- imat=np.linalg.inv(mat)
- print(imat)
- w=np.dot(imat, b)

- w=[6.46689981 0.62064008]

- This model is:

$$\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$$

Multivariate Regression

$$y = w[0] + w[1] * size \qquad y = w[0] + w[1] * d[1]$$

$$y = w[0] + w[1] * d[1] + w[2] * d[2] + w[3] * d[3] + \cdots + w[n] * d[n]$$

Multivariate regression model equation for Rental dataset

$$\text{RENTAL PRICE} = \mathbf{w}[0] + \mathbf{w}[1] \times \text{SIZE} + \mathbf{w}[2] \times \text{FLOOR} + \mathbf{w}[3] \times \text{BROADBAND RATE}$$

Gradient Descent

- There is, however, a simple approach to learning weights that is based on the fact that,
- The error surfaces that correspond to these high-dimensional weight spaces still have that single global minimum.
- This approach uses a guided search from a random starting position and is known as **gradient descent.**

Gradient Descent

- Gradient descent starts by selecting a random point within the weight space .
- Calculate the sum of squared errors(E) associated with this point based on predictions made for each instance in the training set .
- Determine the slope of the error surface $\frac{\partial E}{\partial w[j]}$
- Calculate the value of this derivative at the random point selected in the weight space.

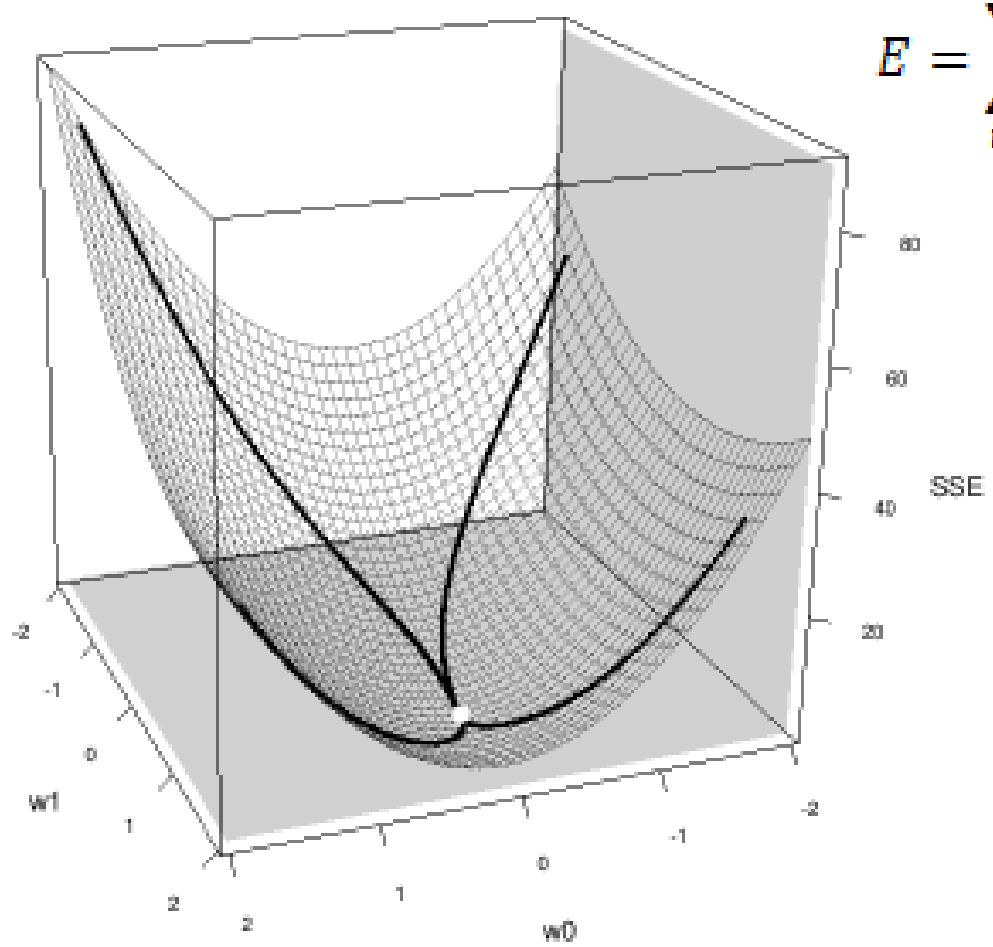
Gradient Descent

- The randomly selected weights are adjusted slightly in the direction of the error surface gradient to move to a new position on the error surface.

- Because the adjustments are made in the direction of the error surface gradient, this new point will be closer to the overall global minimum.

- This adjustment is repeated over and over until the global minimum on the error surface is reached

Gradient Descent



$$E = \sum_{i=1}^{n} (t_i - (w[0] + w[1] * size_i))^2$$

The journey across the error surface that is taken by the gradient descent algorithm when training

the simple version of the office rentals example - involving just SIZE and RENTAL PRICE.

Calculation of best fit weights for the same model by GD

- *1: w - random starting point in the weight space*
- *2: repeat*
- *3: for each w[j] in w do*

$$w[j] \leftarrow w[j] + \alpha \frac{\partial E}{\partial w[j]}$$

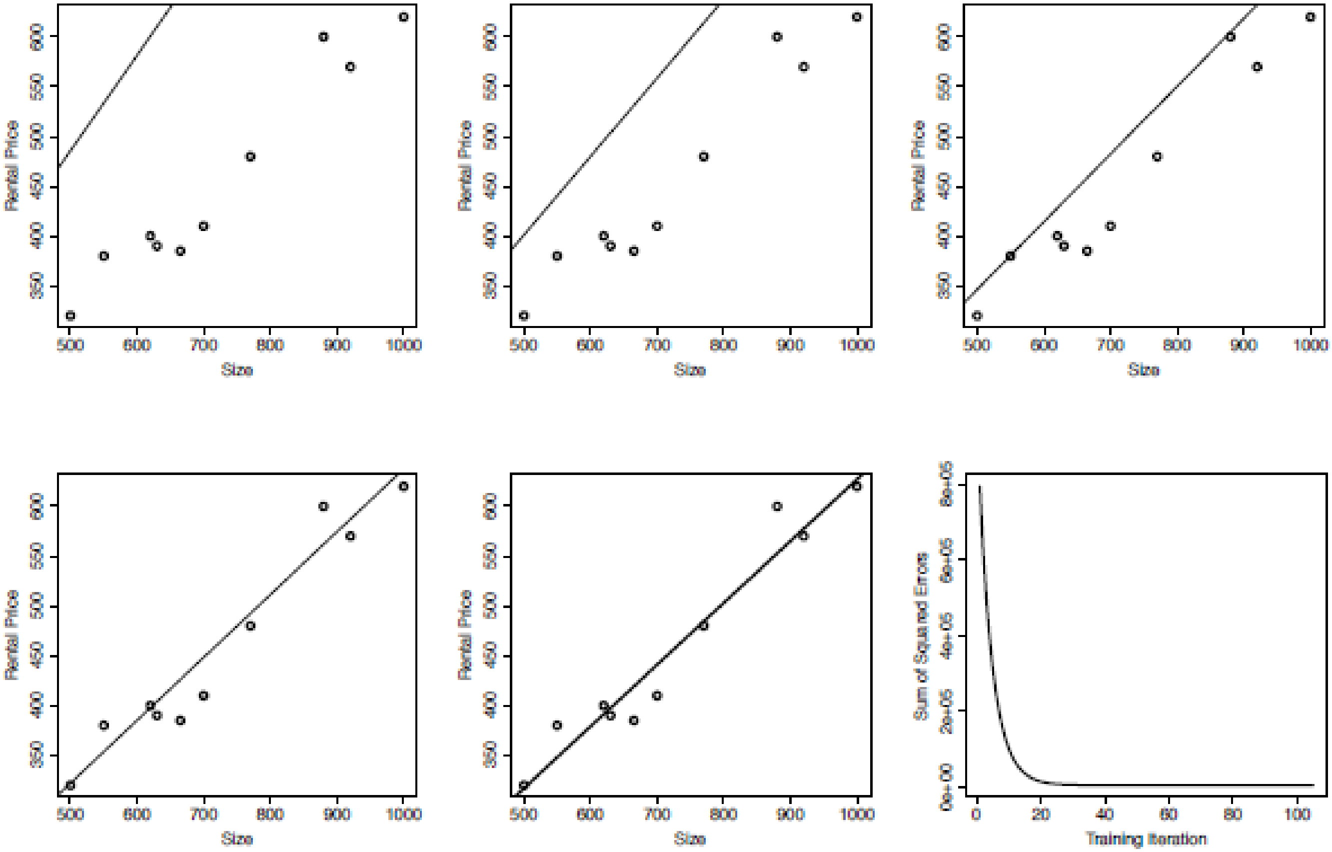
$$w[j] \leftarrow w[j] + \alpha \sum_{i=1}^{n} (t_i - y_i) \frac{-\partial y_i}{\partial w[j]} = \alpha \sum_{i=1}^{n} (t_i - y_i).(-d[j])$$

$$w[j] \leftarrow w[j] + \alpha * errorDelta(d[j], error)$$

- 4: end for
- 5: until convergence occurs

DATA FOLKZ
CATAPULT DATA LEADERS

selection of the simple linear regression models



Multivariate Regression

$$y = w[0] + w[1] * size \qquad y = w[0] + w[1] * d[1]$$

$$y = w[0] + w[1] * d[1] + w[2] * d[2] + w[3] * d[3] + \cdots + w[n] * d[n]$$

Multivariate regression model equation for Rental dataset

$$\text{RENTAL PRICE} = \mathbf{w}[0] + \mathbf{w}[1] \times \text{SIZE} + \mathbf{w}[2] \times \text{FLOOR} + \mathbf{w}[3] \times \text{BROADBAND RATE}$$

A worked Example

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

A worked Example

- For this example let's assume that:
  - $\alpha = 0.00000002$

Initial Weights

w[0]:	-0.146	**w**[1]:	0.185	**w**[2]:	-0.044	**w**[3]:	0.119

A worked Example

Iteration 1

ID	Rental Price	Pred.	Error	Squared Error	errorDelta($\mathcal{D}$, w[i]) w[0]	w[1]	w[2]	w[3]
1	320	93.26	226.74	51411.08	226.74	113370.05	906.96	1813.92
2	380	107.41	272.59	74307.70	272.59	149926.92	1908.16	13629.72
3	400	115.15	284.85	81138.96	284.85	176606.39	2563.64	1993.94
4	390	119.21	270.79	73327.67	270.79	170598.22	1353.95	6498.98
5	385	134.64	250.36	62682.22	250.36	166492.17	2002.91	25036.42
6	410	130.31	279.69	78226.32	279.69	195782.78	1118.76	2237.52
7	480	142.89	337.11	113639.88	337.11	259570.96	3371.05	2359.74
8	600	168.32	431.68	186348.45	431.68	379879.24	5180.17	21584.05
9	570	170.63	399.37	159499.37	399.37	367423.83	5591.23	3194.99
10	620	187.58	432.42	186989.95	432.42	432423.35	3891.81	10378.16
			Sum	1067571.59	3185.61	2412073.90	27888.65	88727.43

A worked Example

$$w[j] \leftarrow w[j] + \alpha errorDelta(d[j], w[j])$$

Initial Weights

w[0]:	-0.146	**w**[1]:	0.185	**w**[2]:	-0.044	**w**[3]:	0.119

Example

$$\mathbf{w}[1] \leftarrow 0.185 + 0.00000002 \times 2,412,074 = 0.23324148$$

New Weights (Iteration 1)

w[0]:	-0.146	**w**[1]:	0.233	**w**[2]:	-0.043	**w**[3]:	0.121

A worked Example

Iteration 2

ID	Rental Price	Pred.	Error	Squared Error	errorDelta($\mathcal{D}$, w[i]) w[0]	w[1]	w[2]	w[3]
1	320	117.40	202.60	41047.92	202.60	101301.44	810.41	1620.82
2	380	134.03	245.97	60500.69	245.97	135282.89	1721.78	12298.44
3	400	145.08	254.92	64985.12	254.92	158051.51	2294.30	1784.45
4	390	149.65	240.35	57769.68	240.35	151422.55	1201.77	5768.48
5	385	166.90	218.10	47568.31	218.10	145037.57	1744.81	21810.16
6	410	164.10	245.90	60468.86	245.90	172132.91	983.62	1967.23
7	480	180.06	299.94	89964.69	299.94	230954.68	2999.41	2099.59
8	600	210.87	389.13	151424.47	389.13	342437.01	4669.60	19456.65
9	570	215.03	354.97	126003.34	354.97	326571.94	4969.57	2839.76
10	620	187.58	432.42	186989.95	432.42	432423.35	3891.81	10378.16
			Sum	886723.04	2884.32	2195615.84	25287.08	80023.74
Sum of squared errors (Sum/2)				443361.52				

A worked Example

$$w[j] \leftarrow w[j] + \alpha errorDelta(d[j], w[j])$$

Initial Weights (Iteration 2)

w[0]:	-0.146	**w**[1]:	0.233	**w**[2]:	-0.043	**w**[3]:	0.121

Exercise

$\mathbf{w}[1] \leftarrow ?, \alpha = 0.00000002$

New Weights (Iteration 2)

w[0]:	?	**w**[1]:	?	**w**[2]:	?	**w**[3]:	?

A worked Example

$$w[j] \leftarrow w[j] + \alpha errorDelta(d[j], w[j])$$

Initial Weights (Iteration 2)

w[0]:	-0.146	**w**[1]:	0.233	**w**[2]:	-0.043	**w**[3]:	0.121

Exercise

$\mathbf{w}[1] \leftarrow ?, \alpha = 0.00000002$

New Weights (Iteration 2)

w[0]:	?	**w**[1]:	?	**w**[2]:	?	**w**[3]:	?

New Weights (Iteration 2)

w[0]:	-0.145	**w**[1]:	0.277	**w**[2]:	-0.043	**w**[3]:	0.123

A worked Example

- After 100 iterations the final values for the weights are:
  - $\mathbf{w}[0] = -0.1513$,
  - $\mathbf{w}[1] = 0.6270$,
  - $\mathbf{w}[2] = -0.1781$
  - $\mathbf{w}[3] = 0.0714$

Prediction

- Using this model:

$$\begin{aligned}\text{RENTAL PRICE} = -0.1513 &+ 0.6270 \times \text{SIZE} \\ &- 0.1781 \times \text{FLOOR} \\ &+ 0.0714 \times \text{BROADBAND RATE}\end{aligned}$$

- we can, for example, predict the expected rental price of a 690 square foot office on the 11^{th} floor of a building with a broadband rate of 50 Mb per second as:

$$\begin{aligned}\text{RENTAL PRICE} &= -0.1513 + 0.6270 \times 690 \\ &\quad -0.1781 \times 11 + 0.0714 \times 50 \\ &= 434.0896\end{aligned}$$

Linear Regression with Scikit

- Load Dataset and split target and Features
- Data Preprocessing(null values, scaling)
- Exploratory Data Analysis
- Split data into training and test part
- Fit the Model to the training data
- Print model coefficients
- Make predictions on the test data
- Compute the mean squared error
- Make Predictions for new query

Assessing Model Using Metrics

The various metrics used to evaluate the results of the prediction are :

1. Mean Squared Error(MSE)
2. Root-Mean-Squared-Error(RMSE)
3. Mean-Absolute-Error(MAE)
4. R^2 or Coefficient of Determination
5. Adjusted R^2

MSE AND RMSE

MSE is simply the average of the squared difference between the target value and the value predicted by the regression model.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)}^{2}$$

The square of the difference between actual and predicted

RMSE is the most widely used metric for regression tasks and is the square root of MSE i.e. the root of averaged squared difference between the target value and the value predicted by the model

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

DATA FOLKZ
CATAPULT DATA LEADERS

R^2 or Coefficient of Determination

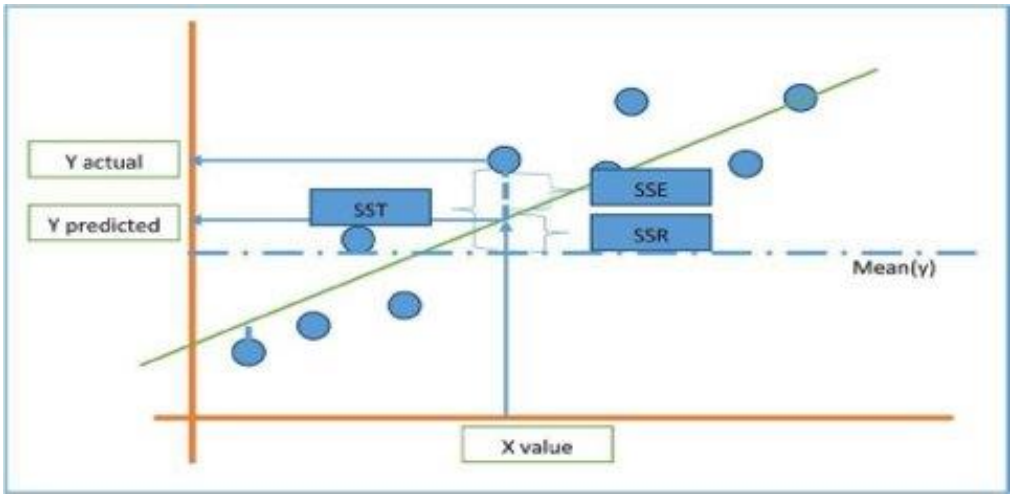
The coefficient of determination(R^2) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable.

It is the ratio of the **variation explained** to the **total variation** (of the dependent variable).

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \Rightarrow 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

$$R^2 = \frac{SS_{\text{reg}}}{SS_{\text{tot}}}$$

DATA FOLKZ
CATAPULT DATA LEADERS



$$SS_{Res} + SS_{Reg} = SS_{Tot}$$

SS_{Res} is the Sum of Square of residuals i.e squared difference between Actual value and the Predicted value

SS_{Reg} is the sum of square regression i.e sum of the differences between the predicted value and the mean of actual value.

SS_{Tot} is the Total Sum of Squares i.e squared differences between the actual value and its **mean**.

The value of R2 lies between 0 to 1. A r-squared value of 1 means the model explains all the variation of the target variable and a value of 0 measures zero predictive power of the model.

$$0 \leq R\text{-}Squared \leq 1$$

The higher the R-square, the better the model.

DATA FOLKZ
CATAPULT DATA LEADERS

Problems with R^2

High R^2 are not always great . The reason for the same is that :

R^2 suffers from the problem that the scores improve on increasing features (independent variables) even though the model is not .

R2 does not provide us with the information regarding the important features for prediction/modelling and it takes all the features while prediction which makes the model biased as every feature may not be useful .

Understanding R2 square problem with an example

Number of washrooms (X1)	Area in acres (X2)	House price in lakhs (Y)
1	0.35	64
2	1	95
1	0.2	55
1	0.75	85
3	1.2	99

Assume , two regression analysis is performed on the data

Regression 1 : Using X2 for predicting Y
R2 – 0.9521
Adjusted R2 – 0.9344

Regression 2 : Using X1 , X2 for predicting Y
R2 – 0.9710
Adjusted R2 – 0.9214

Although number of washrooms is not much useful to predict the price of houses , the R-squared increased from 0.9521 in Regression 1 to 0.9710 in Regression 2 . We may believe that Regression 2 is better model since the R-squared is higher even though the X1 feature is not of much use .

Here , adjusted R-squared comes into picture . The adjusted R-squared checks if the additional feature is important for the model or not . The adjusted R-squared in Regression 1 is 0.9344 compared to the adjusted R-squared in Regression 2 of 0.9214. Therefore, the adjusted R-squared is able to identify that X1 feature is not helpful in predicting the house price . In such a case, based on the adjusted R-squared value Regression 1 model would be selected rather than Regression 2.

Adjusted R^2

To solve the problem of R2 , we have Adjusted R^2 . Adjusted R^2 is an improvement of R^2 .

The Adjusted R-squared measures the variation in the dependent variable, explained by the features which are helpful in making predictions. Unlike R-squared, the Adjusted R-squared would penalize you for adding features which are not useful for predicting the target.

As we increase the useless variables in the data , the Adjusted R2 will decrease and when the relevant variables are increased , the Adjusted R2 will increase .

DATA FOLKZ
CATAPULT DATA LEADERS

Adjusted R^2 Formula

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$R^2_{adj} = 1 - ((1 - R^2)\frac{N - 1}{N - M - 1}$$

As the number of feature increases, the value in the denominator decreases.
- If the R2 increases by a significant value, then the adjusted r-squared would increase.
- If there is no significant change in R2, then the adjusted r2 would decrease.

R^2 is calculated R squared value
N is number of rows
M is number of columns

Adjusted R^2 is always lower than R^2 as it adjusts for the increasing predictors and only shows improvement if there is a real improvement.

DATA FOLKZ
CATAPULT DATA LEADERS

Example of Adjusted R^2

Suppose we have two models :

M1 is the first model using three features A , B and C for predicting Y
M2 is the second model using two features A , B for predicting Y

Given below are the scores of both the models :

	M1 (using A,B,C to predict Y)	M2 (using A,B to predict Y)
R squared value	0.56	0.52
Adjusted R2 value	0.45	0.34

What can you interpret from the above table ?

DATA FOLKZ
CATAPULT DATA LEADERS

Interpretation

Interpretation 1 :
We can see from the values of both R2 and Adjusted R2 that model 1 is better than model 2 as the both the values are higher for Model 1

Interpretation 2 :
From the Adjusted R2 value of the both the models , We can interpret that feature C is important for our model prediction as the Adjusted R2 value is decreasing when we are not including feature C in Model 2

Train Test Split

Before solving any ML problem we should know Train-Test split.

Why to split dataset into Training - Testing data ?

Ok, lets assume you use the **entire** dataset for training the ML model. Then where do we bring data for testing its performance before putting the model to good use .

The simplest way to solve this is to split the data into 2 parts, say 80% for Training the Model and rest 20% for testing the Model.
In this way, we come to know How good or bad is our model. *Like should we **use it as it is** or **tune it** or **discard it**.*

DATA FOLKZ
CATAPULT DATA LEADERS

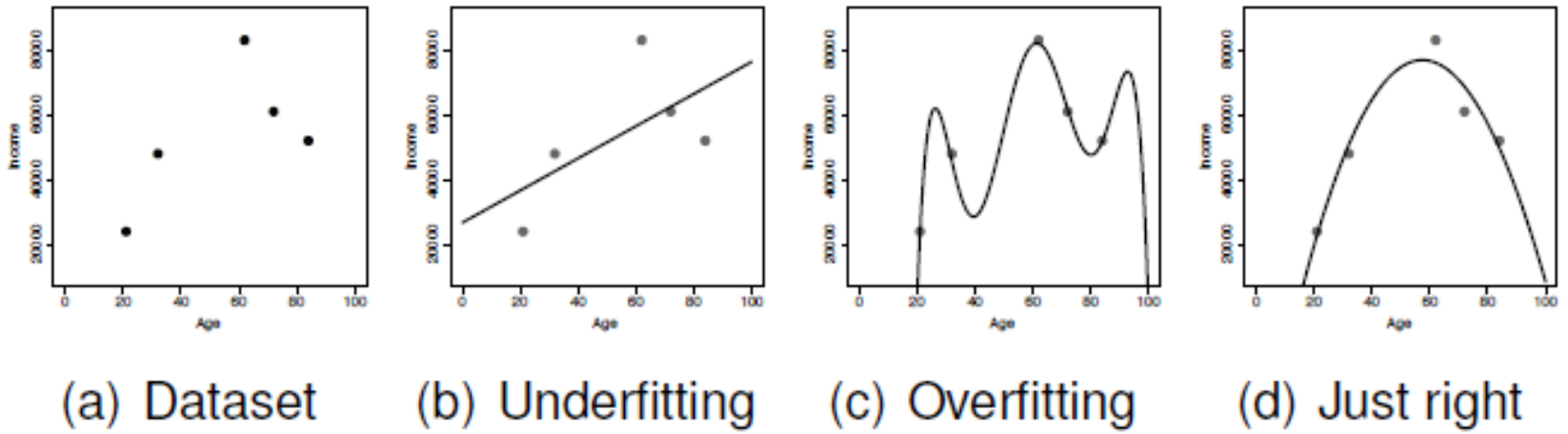
Overfitting and Underfitting

The goal of a good machine learning model is to generalize (Generalisation means how well model behaves on seen and unseen data)well from the training dataset to any dataset from the problem domain.
Model is considered right when it behaves nearly same way on training and test data with highest accuracy.

The poor performance of our model maybe because, the model is too simple to describe the target, or may be model is too complex to express the target.

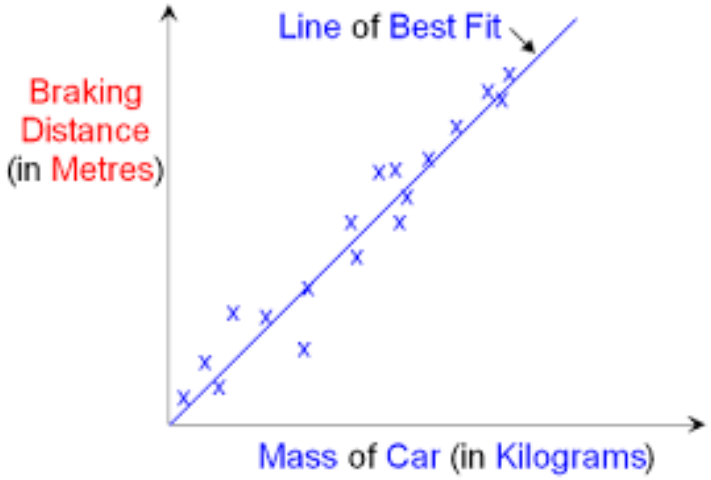
So here comes the concept of overfitting and underfitting.

UnderFitting and OverFitting



- Strike a balance between overfitting and underfitting when trying to predict age from income

What is the Best fit ?



Best fit model is neither Underfit and nor Overfit, it is a generalised model that does not change for seen and unseen data.

Model accuracy should be almost same on training and test data.

DATA FOLKZ
CATAPULT DATA LEADERS

Overfitting and Underfitting example

Our model performed with a 92% accuracy on the training dataset but only 45% accuracy on the test dataset - **Overfitting**

Our model performed with a 45% accuracy on the training dataset but only 42% accuracy on the test dataset - **Underfitting**

Our model performed 92% accuracy on the training dataset and performs 88% - 92% accuracy on the test dataset - **Best Fit**

DATA FOLKZ
CATAPULT DATA LEADERS

Assumptions of Linear Regression

1. **No or little multi collinearity**
2. **Homoscedasticity**
3. **Linear relationship**
4. **No Autocorrelation**
5. **Normal Distribution of error terms**

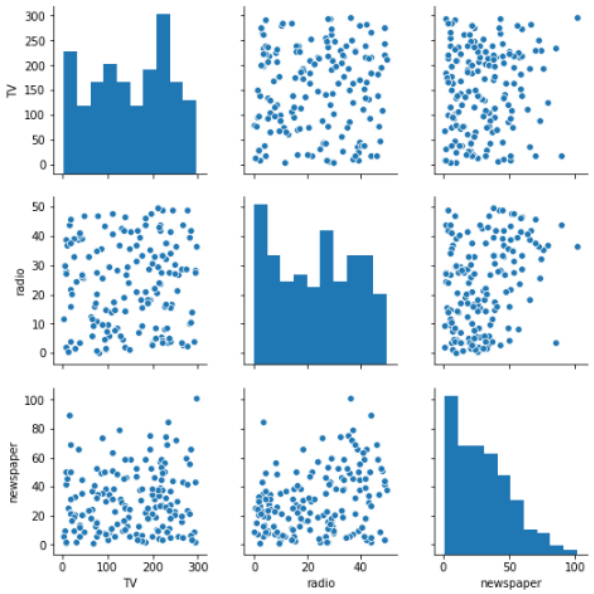
DATA FOLKZ
CATAPULT DATA LEADERS

Assumption 1: No or little multi collinearity .

Multicollinearity is a state of very high inter-correlations or inter-associations among the independent variables.

Multicollinearity exists when:

- One independent variable is correlated with another independent variable.
- One independent variable is correlated with a linear combination of two or more independent variables.



How multi collinearity impacts our model ?

- When independent variables are highly correlated, change in one variable would cause change to another and so the model results fluctuate significantly.

- The Coefficient estimation may not be precise , the errors are likely to be high .

- The unstable nature of the model may cause **overfitting** (overfitting will be discussed in further slides). If we apply the model to another sample of data, the accuracy will drop significantly compared to the accuracy of the training dataset.

Multicollinearity may be checked multiple ways

DATA FOLKZ
CATAPULT DATA LEADERS

	OverallQual	YearBuilt	TotalBsmtSF	1stFlrSF	2ndFlrSF	GrLivArea	PoolArea	MoSold	YrSold	SalePrice
OverallQual	1	0.572323	0.537808	0.476224	0.295493	0.593007	0.0651658	0.0708152	-0.0273467	0.790982
YearBuilt	0.572323	1	0.391452	0.281986	0.0103077	0.19901	0.00494973	0.0123985	-0.0136177	0.522897
TotalBsmtSF	0.537808	0.391452	1	0.81953	-0.174512	0.454868	0.126053	0.0131962	-0.0149686	0.613581
1stFlrSF	0.476224	0.281986	0.81953	1	-0.202646	0.566024	0.131525	0.0313716	-0.0136038	0.605852
2ndFlrSF	0.295493	0.0103077	-0.174512	-0.202646	1	0.687501	0.0814869	0.0351644	-0.0286999	0.319334
GrLivArea	0.593007	0.19901	0.454868	0.566024	0.687501	1	0.170205	0.0502397	-0.0365258	0.708624
PoolArea	0.0651658	0.00494973	0.126053	0.131525	0.0814869	0.170205	1	-0.0337366	-0.0596889	0.0924035
MoSold	0.0708152	0.0123985	0.0131962	0.0313716	0.0351644	0.0502397	-0.0337366	1	-0.145721	0.0464322
YrSold	-0.0273467	-0.0136177	-0.0149686	-0.0136038	-0.0286999	-0.0365258	-0.0596889	-0.145721	1	-0.0289226
SalePrice	0.790982	0.522897	0.613581	0.605852	0.319334	0.708624	0.0924035	0.0464322	-0.0289226	1

1) Correlation Matrix :

This correlation matrix gives us the correlation coefficients of each feature with respect to one another . We can see that, quite a few variables are correlated to each other. There is one pair of independent variables with more than 0.8 correlation which are total basement surface area and first floor surface area. Houses with larger basement area tend to have bigger first floor area as well and so the high correlation should be expected.

2) Variance Inflation Factor (VIF) – VIF is a measure of multi-collinearity in the set of multiple regression variables. The higher the value of VIF the higher correlation between this variable and the rest. VIF values higher than 10 indicate that multicollinearity is a problem.

	features	vif_value
0	OverallQual	50.558204
1	YearBuilt	7015.226148
2	TotalBsmtSF	23.974354
3	1stFlrSF	700.064200
4	2ndFlrSF	141.592671
5	GrLivArea	1141.861313
6	PoolArea	1.046374
7	MoSold	6.529697
8	YrSold	6542.476374

From the results, we can see that most features are highly correlated with other independent variables and only two features can pass the below 10 threshold.

Mathematical Formula of VIF :

$$VIF_i = \frac{1}{1 - R_i^2}$$

R2 is the coefficient of determination .

When R2 is 0 , the VIF value is 1 , that means multicollinearity does not exist in the model .

DATA FOLKZ
CATAPULT DATA LEADERS

How to deal with Multi collinearity

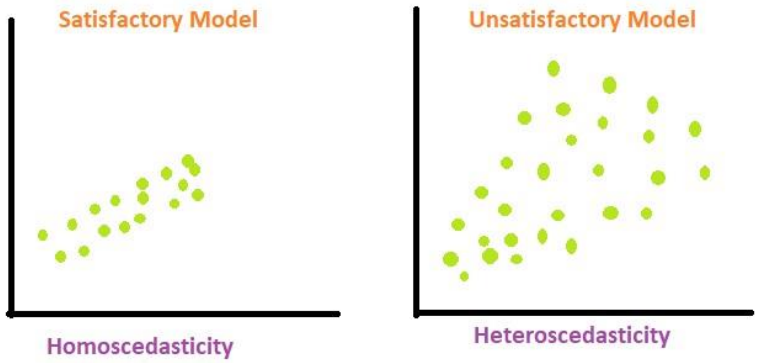
- By Removing some of the highly correlated independent variables.
- By Linearly combining the independent variables, such as adding them together.
- By Performing regularization (will discuss in future sessions)

DATA FOLKZ
CATAPULT DATA LEADERS

Assumption 2 : Homoscedasticity

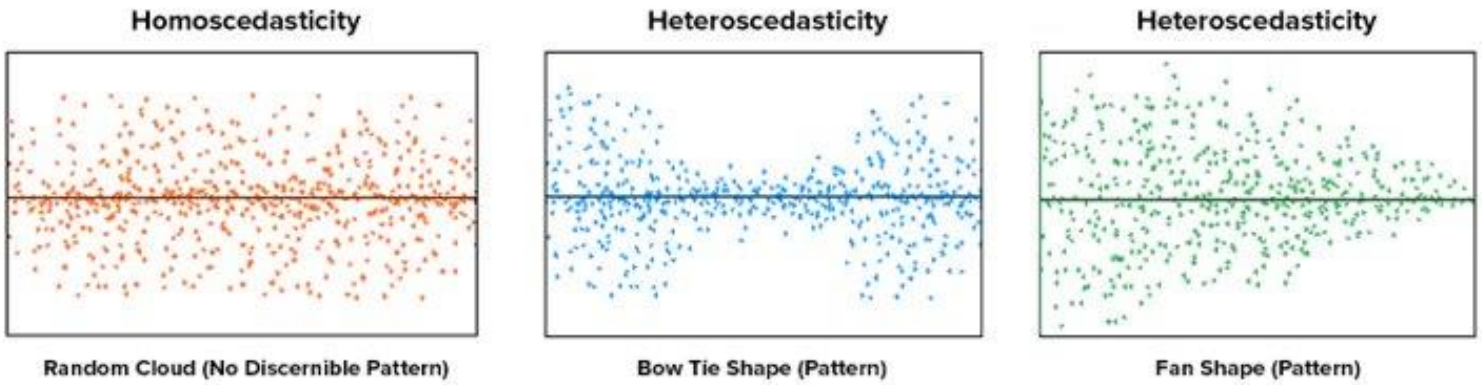
Homoscedasticity describes a situation in which the error term is the same across all values of the independent variables.

Heteroscedasticity (the violation of homoscedasticity) is present when the size of the error term differs across values of an independent variable.



DATA FOLKZ
CATAPULT DATA LEADERS

How to check heteroscedasticity



A scatter plot of **residual values VS predicted values** is a good way to check for homoscedasticity.
In the case of time series data, a plot of **residuals versus time** is plotted
There should be no clear pattern in the distribution and if there is a specific pattern, the data is heteroscedastic.

DATA FOLKZ
CATAPULT DATA LEADERS

Issues caused by violation of Homoscedasticity assumption

- Heteroscedasticity have the effect of giving too much weight to a small subset of the data (namely the subset where the error variance was largest) when estimating coefficients which causes biasness in the calculation amd impacts model performance .

- Heteroscedasticity results in biased standard error . Standard error is central to conducting significance tests and calculating confidence intervals, biased standard errors lead to incorrect conclusions about the significance of the regression coefficients.

DATA FOLKZ
CATAPULT DATA LEADERS

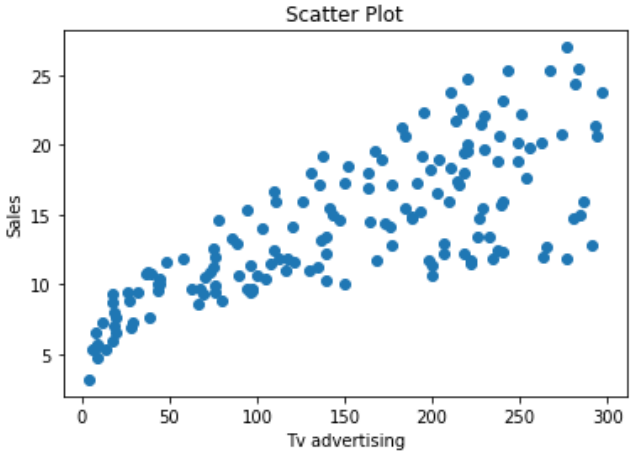
How to fix it ?

- One approach for dealing with heteroscedasticity is to transform the dependent variable using one of the variance stabilizing transformations. A logarithmic transformation can be applied to highly skewed variables.

- In time series models, heteroscedasticity often arises due to the effects of inflation . Some combination of *logging or deflating* will often stabilize the variance in this case.

DATA FOLKZ
CATAPULT DATA LEADERS

Assumption 3 : Linear relationship



Linear regression needs the relationship between the independent and dependent variables to be linear. It is also important to check for outliers since linear regression is sensitive to outlier effects.

The linearity assumption can best be tested with scatter plots.

DATA FOLKZ
CATAPULT DATA LEADERS

Assumption 4 : Normal Distribution of error terms

The error(residuals) should follow a normal distribution .

If the error terms are non- normally distributed, confidence intervals may become too wide or narrow i.e., unstable. This does not help in estimation of coefficients based on cost function minimization.

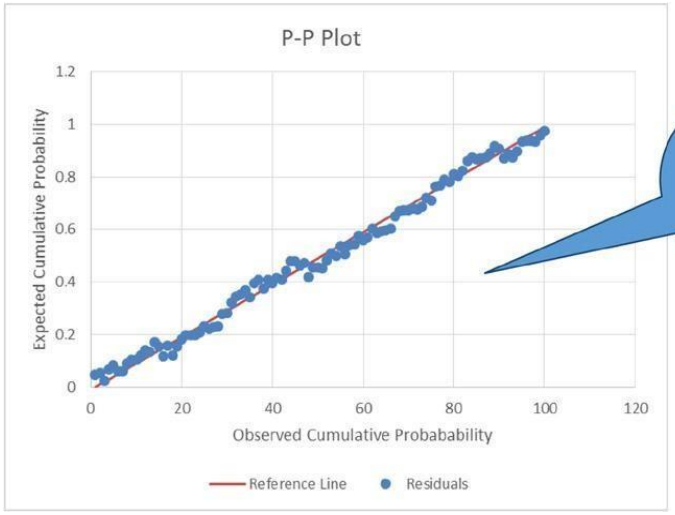
Sometimes the error distribution is "skewed" by the presence of a few large outliers. Since parameter estimation is based on the minimization of *squared* error, a few extreme observations can exert a disproportionate influence on parameter estimates.

Natural log transformation to the variables can be applied to solve this issue .

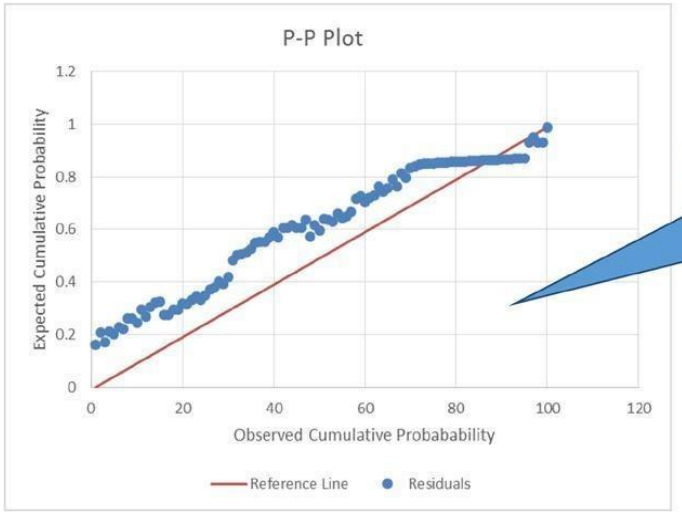
DATA FOLKZ
CATAPULT DATA LEADERS

This assumption can best be tested with histogram or PP plot (Probability Plot) .

Scipy.stats.probplot can be used to plot it in python



Residuals probability distribution is close to normal distribution



Residuals probability distribution is NOT close to normal distribution

DATA FOLKZ
CATAPULT DATA LEADERS

Assumption 5 : No Autocorrelation

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$

Where E_t are residuals from an ordinary least squares regression.

Durbin-Watson test

- Autocorrelation occurs when the residual errors are dependent on each other. The presence of correlation in error terms drastically reduces model's accuracy. The concept of autocorrelation is most often discussed in the context of time series data in which observations occur at different points in time .
- Autocorrelation can be tested with the help of Durbin-Watson test.The null hypothesis of the test is that there is no serial correlation.

DATA FOLKZ
CATAPULT DATA LEADERS

Durbin-Watson test

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$

where the ei = yi – ŷi are the residuals,
n = the number elements in the sample
and
k = the number of independent variables.

- The Durbin Watson statistic is a test statistic used in statistics to detect autocorrelation in the residuals from a regression analysis.
- d takes on values between 0 and 4.
- A value of d = 2 means there is no autocorrelation. A value substantially below 2 (and especially a value less than 1) means that the data is positively autocorrelated, i.e. on average, a data element is close to the subsequent data element.
- A value of d substantially above 2 means that the data is negatively autocorrelated, i.e. on average a data element is far from the subsequent data element.

DATA FOLKZ
CATAPULT DATA LEADERS

Implementation of Linear Regression in Python

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, Y_train)
y_predict = model.predict(X)

rmse = (np.sqrt(mean_squared_error(Y, y_predict)))
 r2 = r2_score(Y, y_predict)
```

DATA FOLKZ
CATAPULT DATA LEADERS