Simple Linear Regression

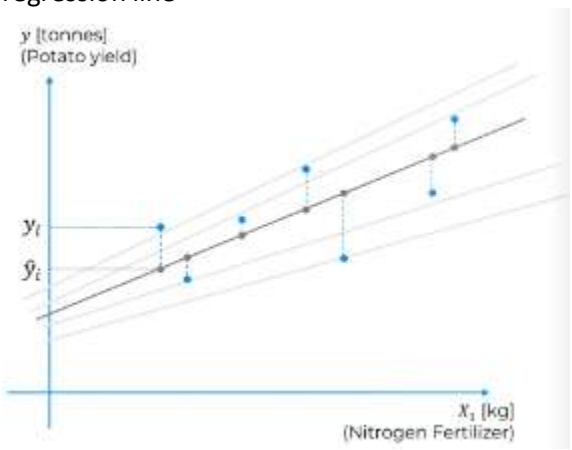
Monday, December 9, 2024 8:46 PM

Used for predicting a real value, like salary for example.

Simple Linear Regression

- Dependent variable = y-intercept + (slope coefficient * Independent Variable)
- $\hat{y} = b_0 + b_1 X_1$

  Dependent variable; y-intercept (constant); Slope coefficient; Independent variable
- This creates a slope

- Which is be best linear regression
  - Ordinary Least Squares
    - Take data points and project them vertically onto the linear regression line(s)
    - You have Yi and Y^, where Yi is the point and Y^ is the point on the regression line
      - The residual: Ei = Yi - Y^
      - b0, b1 such that SUM(Yi - Y^)^2 is minimized
    - Basically we need to take all of the residuals (differences), sqaure them for every data point, add up the sum, and when we get the smallest value, that will be the best regression line
    - 

- The constant in simple linear regression is where the lins crosses the vertical axes

How to do Simple Linear Regression

- Import libraires/dataset
- Split dataset into training and test
- Train linear regression on the training set
  - From sklearn.linear_model import LinearRegression
  - Regressor = LinearRegression()
  - Regressor.fit(feature_train, dependent_train)
- Predict the test set results
  - Dependent_pred = regressor.predict(feature_test)
- Visualize the training set results
  - Plt.scatter(x_train, y_train, color='red')
  - Plt.plot(x_train, regressor.predict(x_train), color='blue')

    - Plt.show()
- Visualize the test set results
    - Plt.scatter(x_test, y_test, color='red')
    - Plt.plot(x_train, regressor.predict(x_train), color='blue')
    - Plt.show()

    - # If the regression line closely aligns with the test data points (red dots), it indicates that the model is making accurate predictions on unseen data.

Multiple Linear Regression

Monday, December 9, 2024 8:47 PM

Multiple Linear Regression

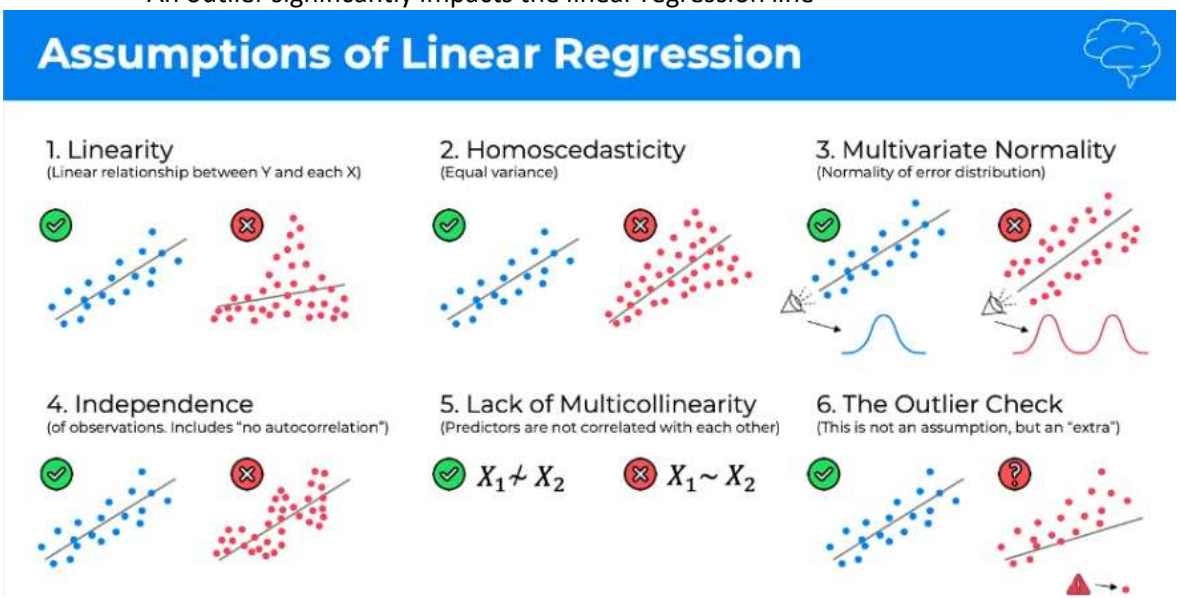
- $$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \cdots + b_nX_n$$

  Dependent variable; y-intercept (constant); Slope coefficient 1; Independent variable 1; Slope coefficient 2; Independent variable 2; Slope coefficient n; Independent variable n

- We might want to predict how many potatoes we harvest by how much fertilizer we are using, how much sun, and the weather

Assumptions of Linear Regression

- Anscombes quartet
  - You cant blindly apply linear regression
  - Need to make sure data set is fit for linear regression
- Assumptions
  - Linearity
    - Linear relationship between Y and each X
  - Homoscedasticity
    - Equal Variable
      - No cone type shape on the chart
  - Multivariate Normality
    - Normality of error distrubution
  - Independence
    - Of observations. Includes 'no autocorrelation'
      - WE don't want to see a pattern
      - Some rows affect other rows
  - Lack of Multicollinearity
    - Predictors are not correlated with each other
  - The Outlier Check
    - An outlier significantly impacts the linear regression line

- 

Multiple Linear Regression Intuition

- Y = b0 + b1*x1 + b2*x2 + b2*x3 + ???
  - We cant add a word into the equation. We can add categorical variables though
  - We need to create dummy variables
- Create a new column for each categorical variable
  - Populate the columns with 1 or 0s
    - DUMMY VARIABLES
- + b4*D1 (this is either 1 or 0)
  - We only want one column at a time from the dummy table
  - Basically swapping from one or two dummy variables

- We don't include both dummy variables because it is the **'dummy variable trap'**
  - Always omit one dummy variable
    - If you have 9, only include
    - If you have multiple dummy tables, do the same thing

Statistical Significance

- Given a coin toss, either heads or tails
  - Two possible alternative universes. Fair and unfair coin
  - Assume the coin is fair
  - When you flip a coin and it lands on 'tails' it has a .5
    - Then tails again is .25, and then .12, …
  - This is the P-Value dropping. The probability of this happening, given that it is a fair universe
  - Statistical Significance is 5%
    - If it is 5% or less we reject the null hypothesis

Building A Model Step-by-Step

- There may be a lot of columns. We may need to decide what columns we want to keep
  - Garbage in, garbage out
- 5 Methods of building models
  - All in
    - Throw in all variables
    - Used for preparing for backward elimination
  - **Backward Elimination**
    - Step 1: Select a significance level to stay in the level (e.g. SL = 0.05
    - Step 2: Fit the full model with all possible predictors
    - Step 3: Consider the predictor with the highest P-Vale
      - If P > SL go to step 4
    - Step 4: Remove the predictor
    - Step 5: Fit the model without this variable
      - Go back to step 3 until all variables have P value less than SL
  - Forward Selection
    - Step 1: Select a significance level to enter the model (e.g. SL = 0.05)
    - Step 2: Fit all simple regression models y ~ Xn select the one with the lowest P-value
    - Step 3: Keep the lowest P-Value, and fit all possible models with one extra predictor added to the ones you already have
    - Step 4: Consider the predictor with the lowest P-Value. If P< SL go back to STEP 3
  - Bidirectional Elimination
    - Step 1: Select SL to enter and to stay (e.g. SLENTER = 0.05, SLSTAY = 0.05)

        - Step 2: Perform the next step of Forward Selection (new variables P < SLENTER to enter)
        - Step 3: Perform ALL steps of Backwards Elimination (old variables P < SLSTAY to stay)
        - Step 4: No new variables can enter and no old variables can exit
    - All Possible Models
        - Step 1: Select a criterion of goodness of fit
        - Step 2: Construct All possible regression models: 2^n-1 total combinations
        - Step 3: Select the one with the best criterion
    - Score Comparison

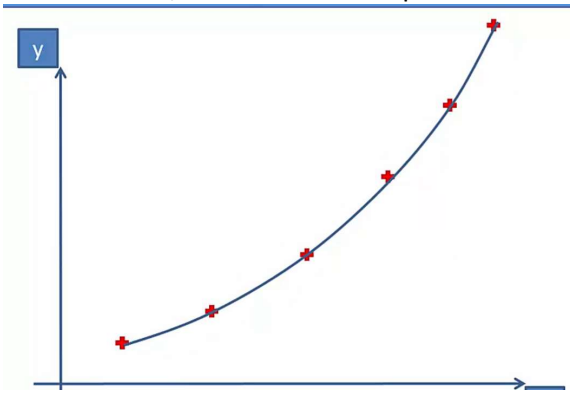
        - Stepwise Regression
            - Backward, Forward, Bidirectional Regression

In multiple linear regression there is no need for feature scaling

Polynomial Regression

Monday, December 9, 2024 8:47 PM

Polynomial Regression

- $y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$
- Same variable, but with different powers
- 

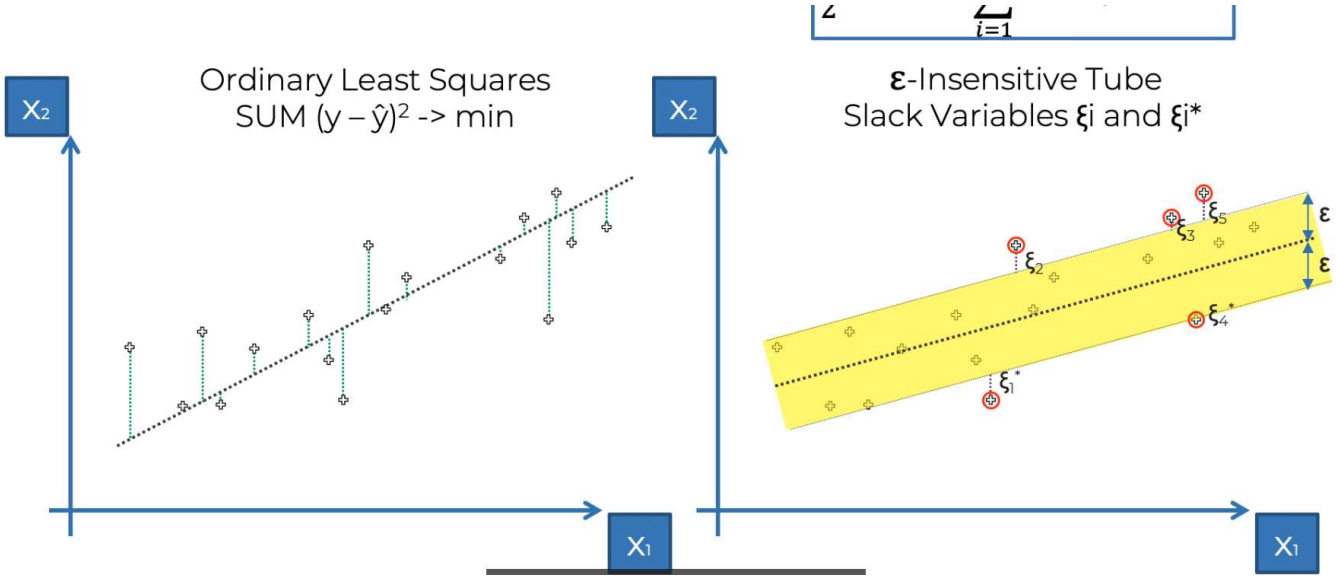
- Called Polynomial 'Linear' Regression because the coefficients are still linear
    - Special case of multiple linear regression
- Used when the dataset represents a curved graph

Support Vector Regression

Monday, December 9, 2024 8:48 PM

Support Vector Regression

- Instead of a line, it will be a tube
  - Has a width of epsolin
    - Any point that are within the tube, will be disregarding error
    - Margin of error we are allowing our model to have
- Has a buffer to our model.
- We also have points outside of the tube, which we do care about the error
  - Slack Variables
  - Ei and Ei*



-

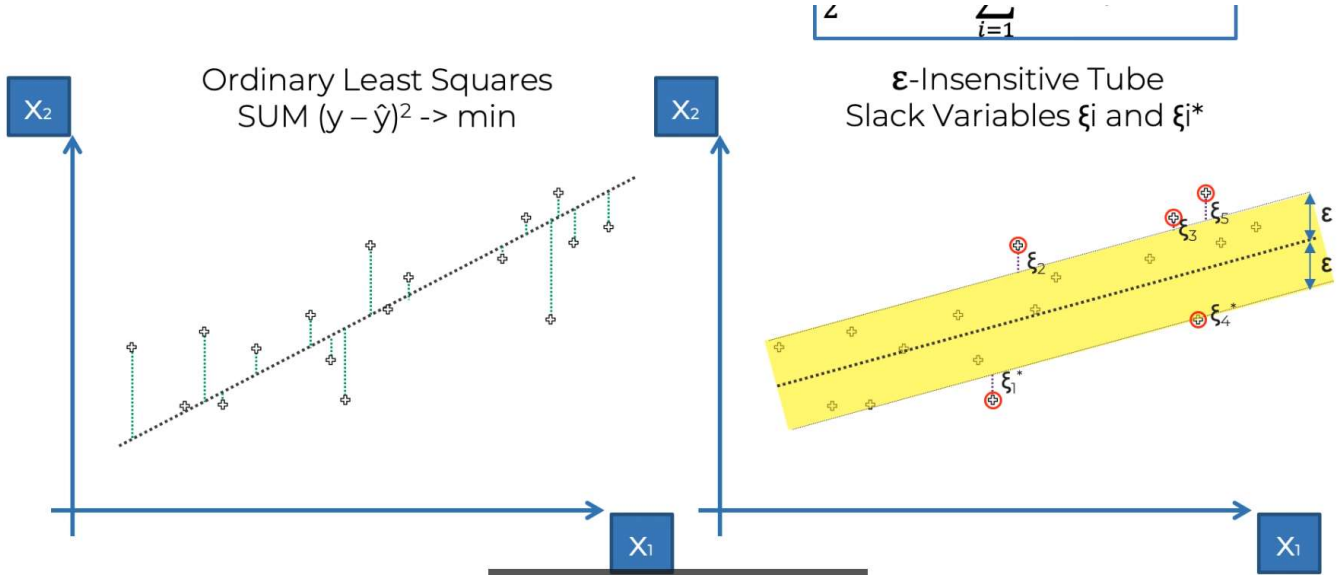
- The slack variables is a vector, support vectors.
  - They support the structure formation of the tubes

Trees/Forests

Monday, December 9, 2024 8:48 PM

Support Vector Regression

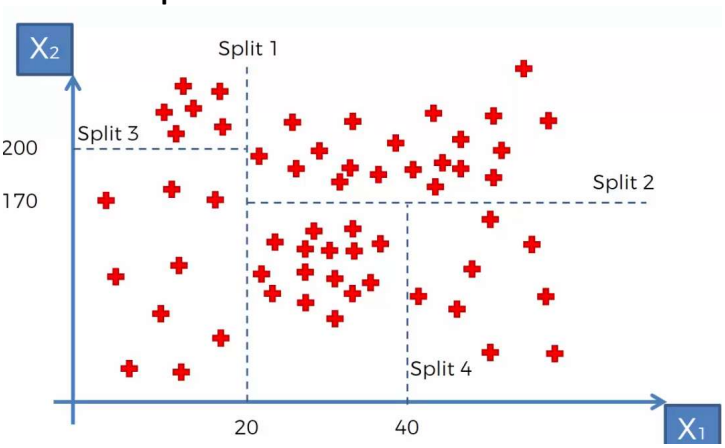
- Instead of a line, it will be a tube
  - Has a width of epsilon
    - Any point that are within the tube, will be disregarding error
    - Margin of error we are allowing our model to have
- Has a buffer to our model.
- We also have points outside of the tube, which we do care about the error
  - Slack Variables
  - Ei and Ei*

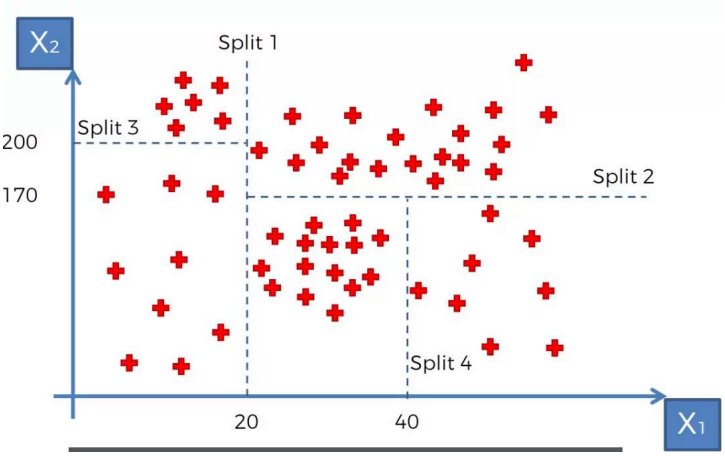


- The slack variables is a vector, support vectors.
  - They support the structure formation of the tubes

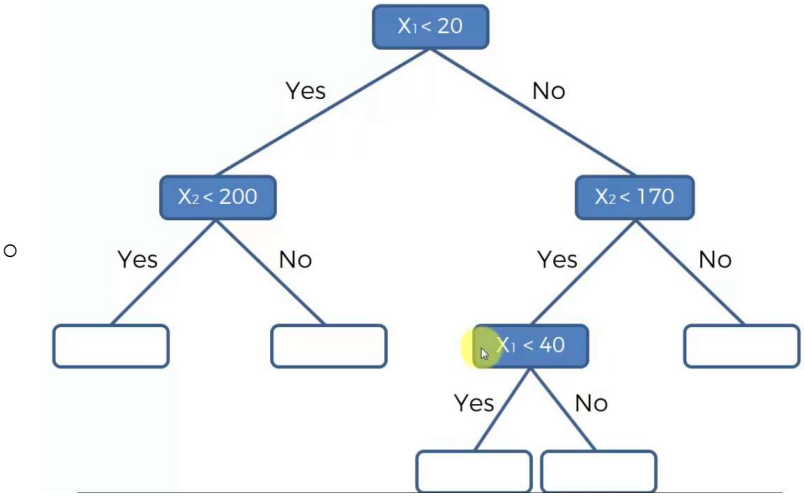
Decision Trees

- CART
  - Classification and Regression Trees
- We have X1 and X2, which are independent variables
  - Scatter plot will be split up into segments
    - May be along the X1 or X2
  - Splits are determined by an algorithm
    - **Splits are called leaves**

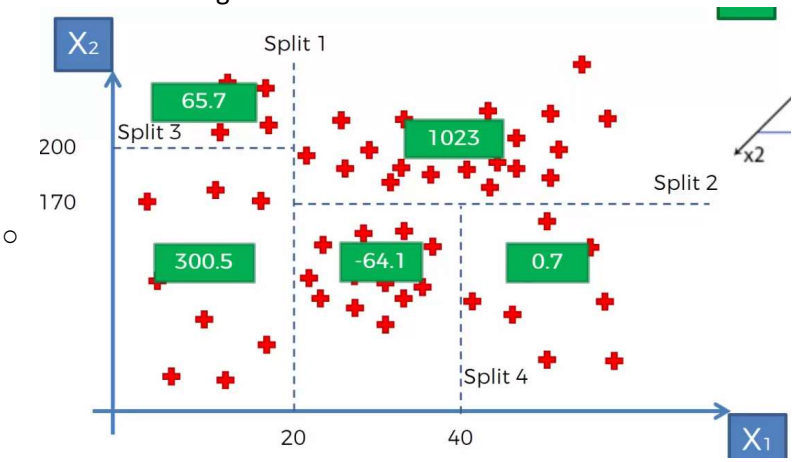




- Create decision tree based on our first split



- What goes in those blank boxes?
  - How do we predict y in that terminal leaf
  - Take the averages of each of the terminal leaves



Regression Decision tree is not really adapted to simple data sets with one feature and one dependent variable

- You also don't need to apply feature scaling
  - This is because the model is expecting successful splits of the data, and not a mathematical formula

Random Forest

- Ensemble Learning
  - Take multiple algorithms and put them together to make them more powerful

- Step 1: Pick at random K data points from the training set
- Step 2: Build the Decision Tree associated to these K data points
- Step 3: Choose the number Ntree of trees you want to build and repeat STEPS 1 & 2
- Step 4: For a new data poinnt, make each one of your nTrees predict the value of Y to for the data point in question, and assign the new data point the average across all of the predicted Y values

- Basically build a bunch of trees from little bits of the data set, run the prediction, and then take the average between all of the trees

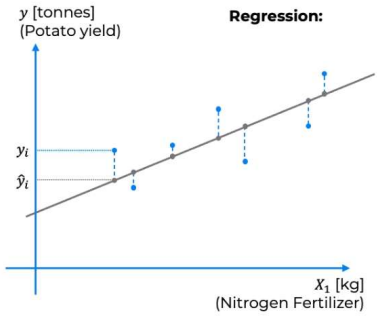
R Squared

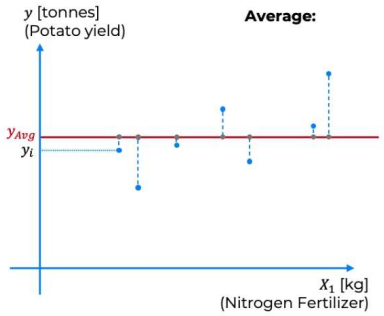
Monday, December 9, 2024 8:49 PM

R Squared

- SS_{res} = SUM(Yi - Y^i)^2
  - Residual Sum of Squares
- SS_{tot} = SUM(Yi - Yavg)^2
  - Total Sum of Squares

- R^2 = 1 - (SSres / SStot)
-





$$SS_{res} = SUM(y_i - \hat{y}_i)^2 \qquad SS_{tot} = SUM(y_i - y_{avg})^2$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- R^2 is between 0 and 1
  - 1.0 is a perfect fir (suspicious)
  - ~0.9 is very good
  - <0,7 is not great
  - <0.4 is terrible
  - <0 model makes no sense for this data
- Basically, greater is better

Adjusted R-Squared

- Problem with R squared
  - If you add a new column to the dataset, the total SS does not change
  - Residual SS will decrease or stay the same
    - This is because we are using Ordinary Least Squares. Aims to minimize the Residual SS
- Solution: Adjusted R^2

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R^2 – Goodness of fit
(greater is better)

Problem:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 \leftarrow + b_3X_3$$

$$SS_{res} = SUM(y_i - \hat{y}_i)^2$$

- SS_{tot} doesn't change
  SS_{res} will decrease or stay the same *(This is because of Ordinary Least Squares: SS_{res}-> Min)*

Solution:

$$Adj\ R^2 = 1 - (1 - R^2) \times \frac{n-1}{}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R^2 – Goodness of fit
(greater is better)

Problem:

$\hat{y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3$

$$SS_{res} = SUM(y_i - \hat{y}_i)^2$$

- SS_{tot} doesn't change
  SS_{res} will decrease or stay the same *(This is because of Ordinary Least Squares: SS_{res}-> Min)*

Solution:

$$Adj\ R^2 = 1 - (1 - R^2) \times \frac{n-1}{n-k-1}$$

- K is the number of independent variables
- N is the sample size

- This new formula penalizes for adding new data

How to find the best model?

- Try all of the models and select the best one
  - Adjusted R^2