Building regression models with scikit-learn

In regression analysis, we typically establish a relationship between one or more independent variables (X) and a dependent variable (Y). The goal is to understand how changes in X can explain or predict changes in Y.

Linear regression:

- Two sets of statistical tools:
  - Descriptive statistics: Used to identify important elements in a data set.
    - Find the dot.
    - Summarize and quantify the data set such as find the mean, mode, median, the range, the interquartile range, variance and so on.
    - measures of central tendency and measures of dispersion.
  - Inferential statistics: Explain the important elements that were identified.
    - Connect the dot. Draw insights via relationships.
    - What are relationships with other elements.
    - Hypothesis testing of proposed explanations.
    - Data modeling.
- Linear Regression model finds the best fitting straight line to the data, and this straight line is our linear model.
- Linear regression is a statistical method used to model the relationship between variables, specifically finding the best-fit linear line between independent and dependent variables.
- Equation of line, Y= A + BX, we need to find the value of A and B such that the line fits best to the data and minimize the R^2.

Minimize the least square error.

- The goal is to minimize the distance between the data point and the line that fits the data (model), that is minimize the sum of the squared differences (residuals) between the actual values and the predicted values.
- R^2 is a measure of how well the linear regression fits the underlying data. How much of the variance in the data points is captured by the linear relationship.

Assumptions of Linear Regression:

1. **Zero Mean Residual:** Ideally, the residuals should have a mean of zero. This means that, on average, the model neither underestimates nor overestimates the actual values.
2. **Constant Variance of residual(Homoscedasticity):** The variance of the residuals should be roughly constant across all levels of the independent variable(s). In other words, the spread of residuals should not systematically change as the values of X change.
3. **Independence of Residuals:** Residuals should be independent of each other. The error in predicting one data point should not depend on the errors in predicting other data points.
4. **Independence of Residuals from Independent Variables (No Autocorrelation):** Residuals should not exhibit any pattern or correlation with the independent variables. There should be no systematic relationship between residuals and X.
5. **Normal Distribution of Residuals:** Residuals should follow a roughly normal distribution. This assumption is often required for hypothesis testing and confidence intervals.

The risks of simple regression can be mitigated by analyzing R^2 and residuals.

Multiple Regression:

- It is a statistical technique used to analyze the relationship between a dependent variable (target) and multiple independent variables (features or predictors).
- It's an extension of simple linear regression, where there's only one independent variable.
- In multiple regression, we use multiple predictors to model the dependent variable.
- The general form of a multiple regression model is:

  $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$

  Where:
  - Y is the dependent variable.
  - β_0 is the intercept.
  - $\beta_1\ \beta_2 \ldots\ldots \beta_p$ are coefficients for the independent variables $X_1, X_2, \ldots, X_p$
  - ϵ is the error term.

Risks in Multiple regression:

1. ***Multicollinearity:*** Multicollinearity occurs when two or more independent variables in your regression model are highly correlated with each other.

   Mitigation:
   - Consider removing one of the highly correlated variables.
   - Standardize the variables.
   - Use dimensionality reduction techniques like Principal Component Analysis (PCA) to extract significant features of data.

2. ***Overfitting:*** It occurs when your regression model is too complex, fitting the training data very closely but performing poorly on new, unseen data.

   Mitigation:
   - Choose an appropriate model complexity (e.g., by selecting the right degree for polynomial regression).
   - Use regularization techniques like Ridge or Lasso regression.

3. ***Underfitting :*** It occurs when your regression model is too simple to capture the underlying relationships in the data.

   Mitigation:
   - Choose a more complex model or feature engineering.
   - Consider using polynomial regression or other nonlinear regression models.

4. ***Model evaluation:*** Evaluating the performance of a multiple regression model is essential to ensure that it provides meaningful insights and accurate predictions.

   Mitigation:
   - Use cross-validation, validate against different datasets, or explore more advanced evaluation techniques depending on your specific problem.

R^2

- R^2 is common and popular metrics for evaluating a regression analysis.
- R^2 is a measure of how well the regression model has captured the variance in the underlying data i.e., how well the straight-line relationship fits the underlying data.
- R^2 is expressed between 0 and 100% or between zero and one.
- 0 – means the model does not explain any variance.
- 1 – means the model explains all the variance.

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$, square of r gives us R^2.

For r,

- The resulting value of r will be between -1 and 1.
- -1 indicates a perfect negative linear relationship.
- 0 indicates no linear relationship.
- 1 indicates a perfect positive linear relationship between the observed and predicted values.

- Higher R^2 indicated better fit, the calculation is simple and involves the correlation between observed and predicted values.

Disadvantages:

- As we add new X variable for multiple regression, we find that the value of R^2 constantly increases, while it can lead to overfitting of the model on the data.

Adjusted R^2

- Adjusted R^2 is preferred for evaluating multiple regression.
- It is derived from R^2, and it basically adds in a penalty for adding irrelevant variables to the regression model.
- It adjusts R^2 by taking the no of independent variables(predictors) into account.
- The adjusted R^2 is always lower than R^2 when there are multiple predictors in the model.

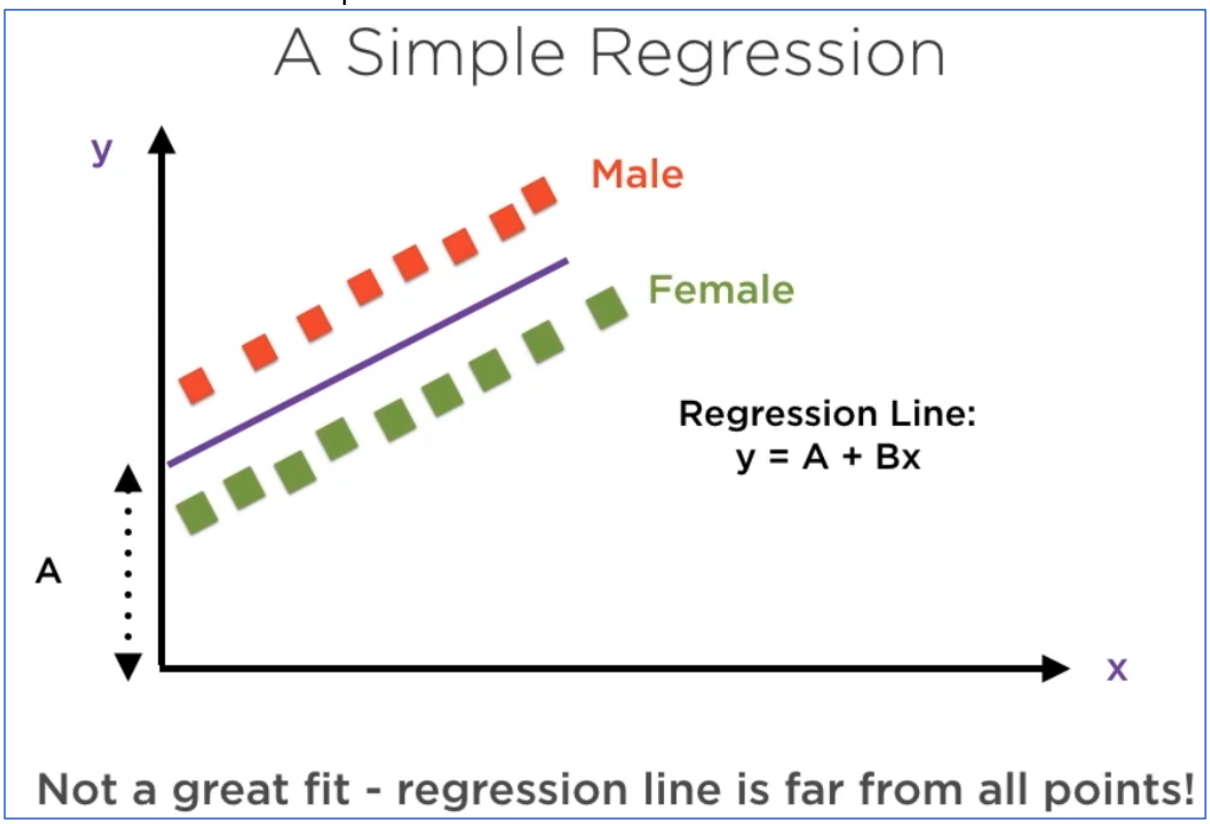
$$\text{Adjusted } R^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-p-1}$$

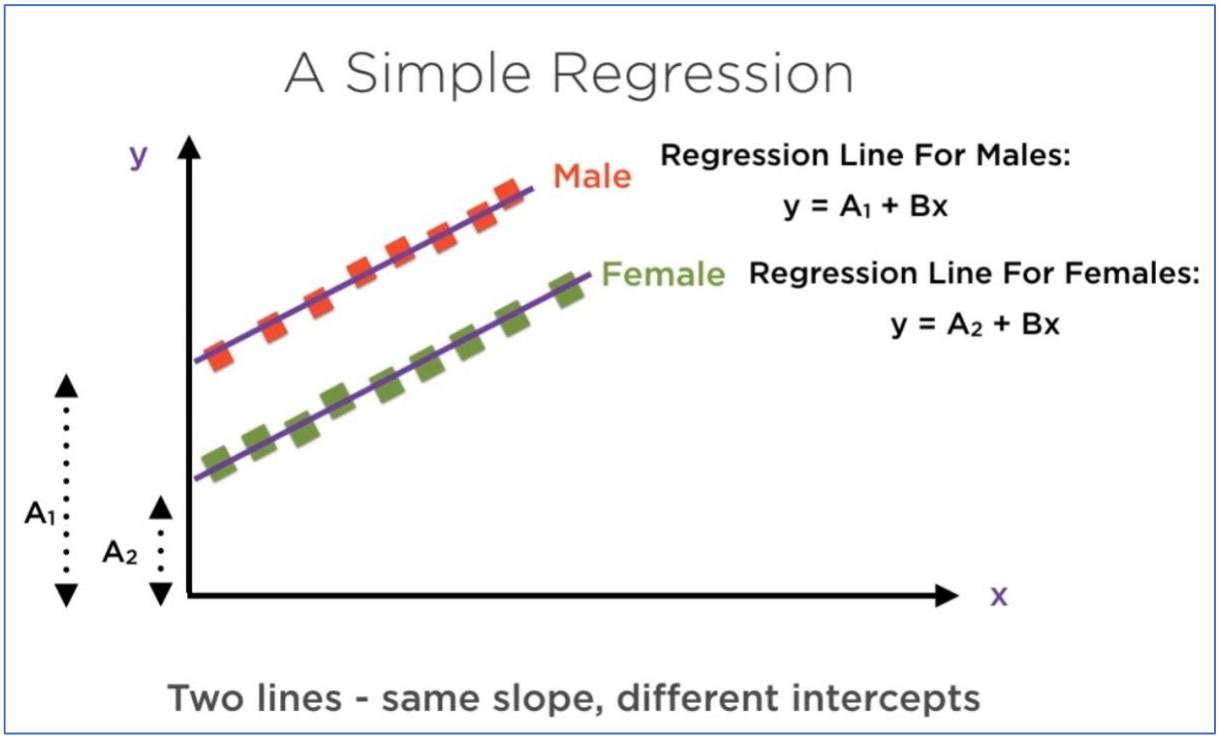
Where:

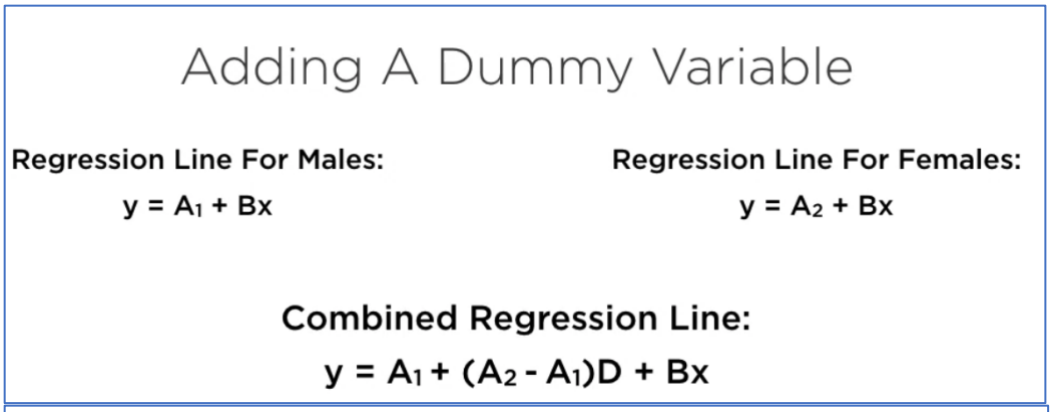
- $\text{Adjusted } R^2$ is the adjusted R-squared value.
- R^2 is the ordinary R-squared value.
- n is the number of data points.
- p is the number of predictors (independent variables) in the model.

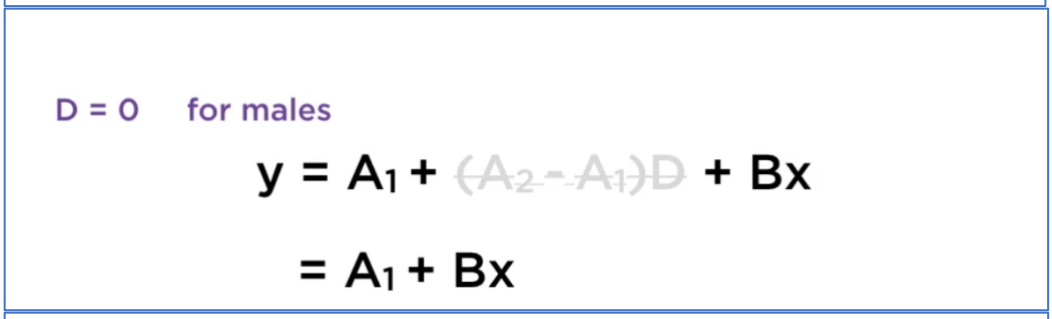
Regression with categorical variables

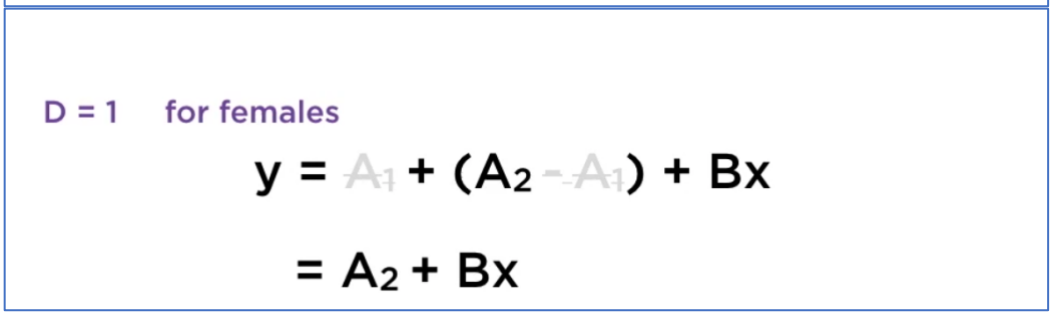
- When there are two categories and two lines to represent the regression, we can introduce dummy value to reduce two lines to one equation.





- 

- 

- 

- This works for data with two categories,
- For data with multiple categories, we set up k-1 dummy values.
- Scikit learn takes care of these.

Regularized Regression:

Overfitting and regularization to mitigate overfitting:

- Overfitting: complex model.
  - Works well for training data and worst on test data.
- Ways to mitigate overfitting:
  - **Regularization**: by penalizing the complex model.
  - **Cross validation**: to build good morning you'll have distinct training and validation phases before you test the model on new data.
  - In neural network models - **Dropout** – this technique is used. where we intentionally turn off some neurons during the training phase.
  - Early stopping
  - Ensembling.
  - Train with more data.

Regularization:

- Penalize the complex model.
- Add penalty to the objective function, like min least squared error in linear regression.
- Penalty term is added, as function of regression coefficient.
    - Higher and more complex regression coefficients have higher penalty.
- Forces optimizer to keep it simple.
- It reduces variance error: the variance error means that the model will be less sensitive to the training data, it won't try to fit the training data very closely.
    - i.e., regularization is often used to reduce the variance in a model.
    - High variance: The model is too sensitive to the training data and fits it too closely, potentially overfitting.
    - Regularization adds penalties to prevent the model from becoming too complex, reduces the variance.
- Regularization increases the bias-variance trade-off of the model.
    - By adding penalty, the regularization prevents the model from becoming complex, which leads to high bias and low variance.
    - Bias is the assumptions/simplification that the model uses to predict outputs.
    - Models with high bias make strong assumptions and tend to be overly simplistic.
    - Low-bias models are more flexible and make fewer assumptions.
- Variance – Bias trade off: we need the model at low bias and low variance. That is a balanced scenario.
    - A well-balanced model is one that captures the underlying patterns in the data without being too simple (high bias) or too complex (high variance), resulting in better generalization to new data.

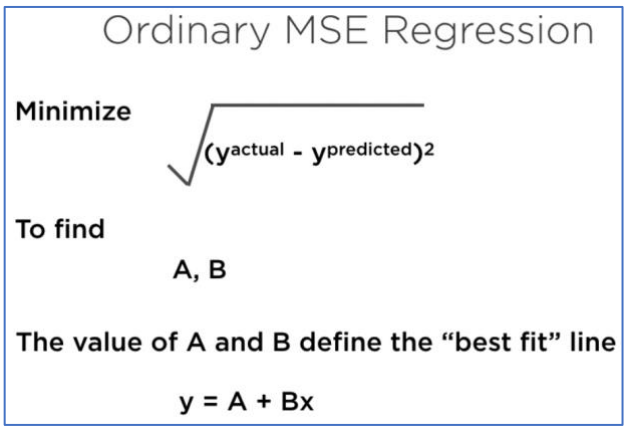
In a nutshell:

Bias	Variance
High bias: simple model, underfitting. Low bias: complex model, overfitting	High variance: complex model, overfitting Low variance: simple model, underfitting

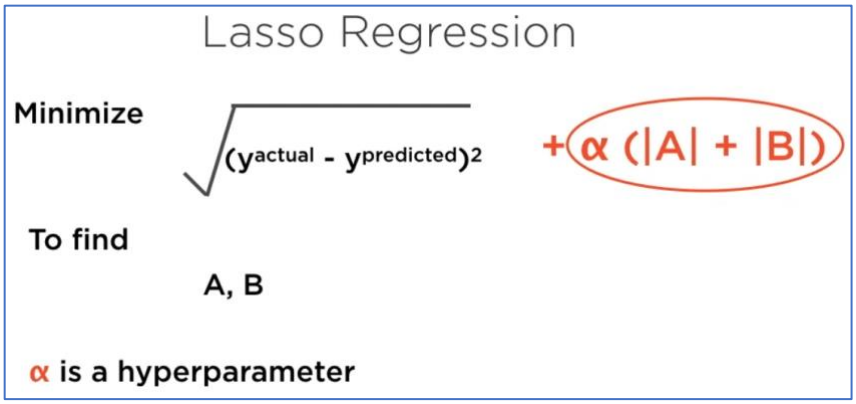
- The simplicity may cause the model to miss important patterns in the data.
- The complex model memorizes the training data, including its noise without capturing the pattern.
- Balance is necessary. And that's variance-bias trade off.

Lasso Regression – L1 norm:

- Lasso regression and Ridge regression penalizes large regression coefficients.
- Elastic net regression is simply a combination of lasso and ridge.
- The difference between these models is the penalty function.
-



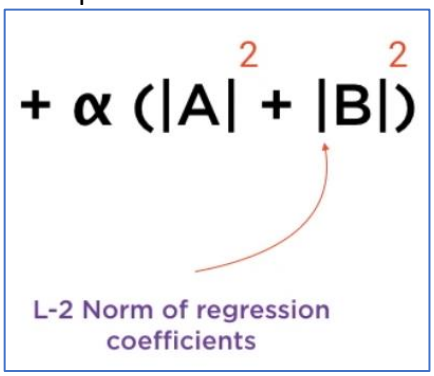
-



- Lasso encourages the model to reduce the absolute values of *less important features to zero*, effectively performing feature selection.
- Therefore, Alpha *eliminates* unimportant features.
- It may introduce *more bias, but lower variance* compared to Ridge when dealing with high-dimensional data.
- More sensitive to outliers, since it can reduce it to zero.

Ridge Regression – L2 norm:

- It is square of coefficients instead of absolute value of coefficient.



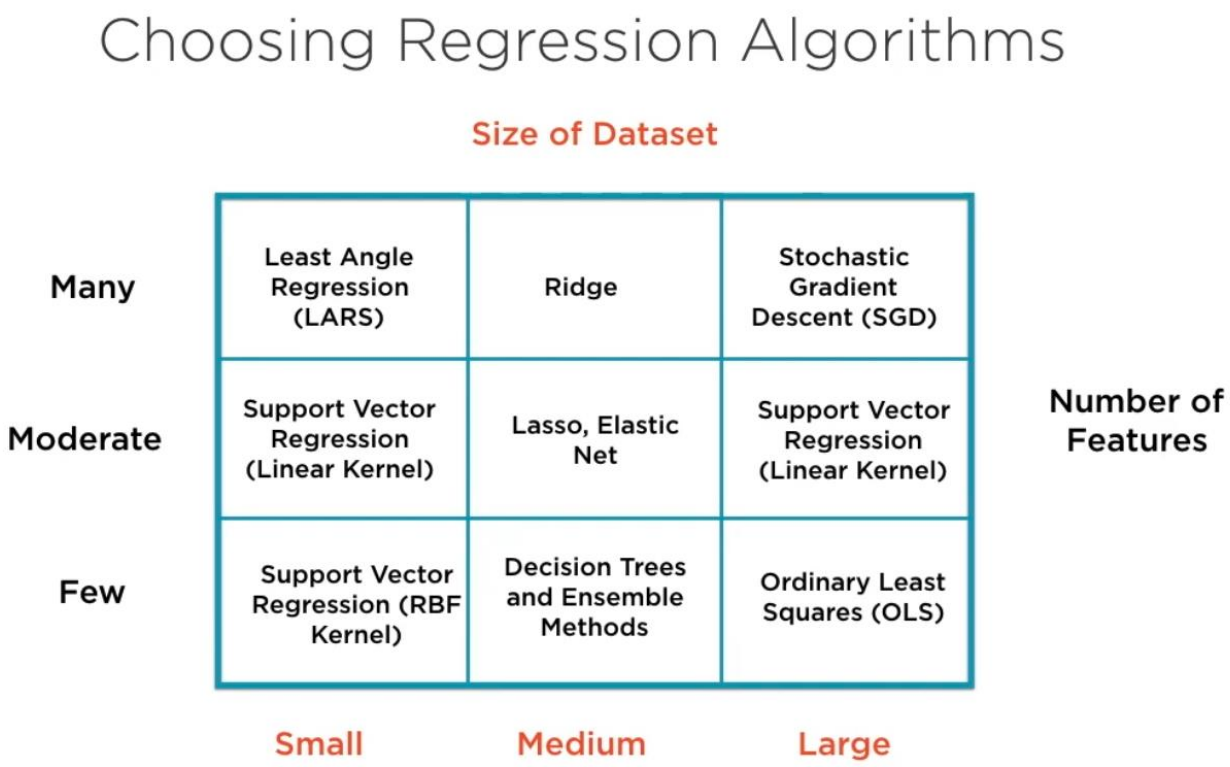
- Ridge shrinks the coefficients of all features but *does not force any of them to become exactly zero*.
- Alpha reduces the magnitude of feature and does not eliminate completely.
- It typically introduces *less bias* than Lasso but may not perform feature selection as effectively.
- Ridge is more robust to outliers. As it can shrink the effect without eliminating.

Elastic Net Regression:

- Combines the strength of lasso and Ridge regression.
- ENR = MSE + Lasso + Ridge
- Lasso and ridge have their own penalty values, alpha 1 and alpha 2.
- Use *cross validation* to find best alpha values for lasso and ridge.
- If the alpha for both is 0, we get cost function itself.
- When alpha 1 is 0, we get ridge regression.
- When alpha 2 is 0, we get lasso regression.
- It does better job dealing with correlated parameters, i.e., to address the multicollinearity.
- In scikit learn we have a library to calculate ENR,

```
1 / (2 * n_samples) * ||y - Xw||^2_2
+ alpha * l1_ratio * ||w||_1
+ 0.5 * alpha * (1 - l1_ratio) * ||w||^2_2
```

Or simply,
1/2N * (MSE) + (alpha * lambda * |coefficients|) + (alpha * (1 – lambda) * |coefficients|^2)
When lambda is 0, Ridge regression.
Lambda is 1, lasso regression.



Support Vector Regressor:

SVM classifier	SV regressor
Find the widest margin between support vectors	Find the best fit line for the points
No points inside margin	Try to maximize the no of points inside the margin.
Points far from margin are good, It improves objective function value.	Points far from margin is bad, It reduces the objective function value.
Outliers on wrong side of margin are penalized.	Points far from the margin are penalized
Width of the margin is found by optimizer, It optimizes the model by finding the widest margin.	Width of the margin is specified beforehand in model. It is a hyperparameter : ε

Nearest Neighbors regression:

- NNR uses the training data to find what is most similar to current sample.
- Find the distance between points using various distance matric available like in classification and pick the closest one.
- NNR techniques:
    - KNN Reg – average value of k near neighbors, and this average is the reg value for new data point.
    - Radius NN reg – average of k neighbors within the radius.

Stochastic Gradient Descent regressor:

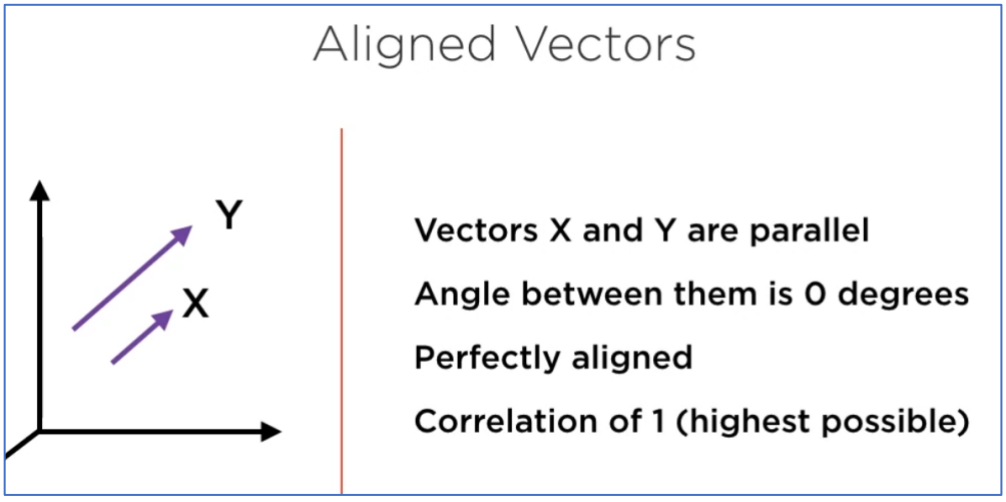
- Iteratively converges to best model that gives best fit line.
- Find the line that minimizes MSE.
- Find reg coefficients that gives min value for MSE.
- Working is similar to classification models.
- We can use different loss function.
- MSE yields OLS regressor.
- Can implement Lasso, Ridge, Elastic Net.
- Works well for large dataset.

Decision Tree regression:

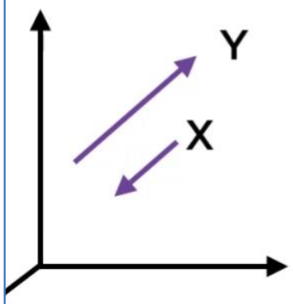
- Make decision based on rules.
- Similar to classification steps.
- Each rule has a threshold.
- Order of decision variables matter.
- Rules and order are found using ML.
- It is referred to as CART.
- DT is used to predict continuous value instead of label.

Least angle regressor:

- It is technique that relies on selecting x-variables that have the highest correlation(least angle) with the unexplained y-variable.
- When there are many features than sample.
- Intuitive and stable.



Opposite Vectors



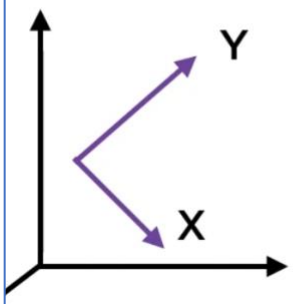
Vectors A and B point in opposite directions

Angle between them is 180 degrees

Perfectly opposed

Correlation of -1 (lowest possible)

Orthogonal Vectors



Vectors X and Y are at 90 degrees

Orthogonal vectors represent uncorrelated data

X and Y are unrelated, independent

Pseudo code:

LARS Regression

```
Start with all coefficients β equal to zero

Find predictor Xj most correlated with y

Increase coefficient of βj
  Until:
  Some other Xi is more highly correlated than Xj

Increase coefficient of βi , βj

Continue until all X variables are in model
```

Example

1. You start with an empty model, which means you have no predictors included yet.
2. LARS examines the correlation between the response variable (final exam score) and each predictor variable (hours of study and practice tests). It selects the variable that has the highest correlation with the current residual, let's say "hours of study."
3. LARS moves the coefficient of "hours of study" toward its least-squares coefficient, effectively fitting a linear regression model with this single predictor.
4. If, at a later step, "practice tests" becomes equally correlated with the current residual, LARS will include it and move the coefficients of both predictors simultaneously, maintaining their correlations.
5. The algorithm continues this process until it reaches the stopping criterion. Suppose you set the criterion to include only two predictors in the model. Once that condition is met, LARS stops.

Reference:

1. Sowjanya Sadashiva: medium blog on Regression and Regularization: https://medium.com/@sowjanyasadashiva/overfitting-and-underfitting-318074b5cf93
2. Plural sight: regression models with scikit learn by Janavi Ravi.
3. Classification notes for reference: https://github.com/SowjanyaSadashivu/Scikit-Learn/blob/f031eedb38dcdac4f86781da5f18f71b458db34c/Building%20classification%20models%20with%20scikit-learn/Building%20Classification%20models%20with%20scikit.pdf