Regressions

IST 347
Dr. Samir Chatterjee

Agenda

- When do we use regressions?
- Simple Linear regression
- Multiple linear regressions
- Working with Python and scikit-learn

About regressions

- Regression analysis is used when you want to predict a continuous dependent variable from a number of independent variables. If the dependent variable is dichotomous, then *logistic regression* should be used
- Use regression analysis to describe the relationships between a set of independent variables and the dependent variable. Regression analysis produces a regression equation where the coefficients represent the relationship between each independent variable and the dependent variable. You can also use the equation to make predictions.

A Researcher View

- For example, imagine you’re a researcher studying any of the following:
  - Do socio-economic status and race affect educational achievement?
  - Do education and IQ affect earnings?
  - Do exercise habits and diet effect weight?
  - Are drinking coffee and smoking cigarettes related to mortality risk?
- All these research questions have entwined independent variables that can influence the dependent variables. How do you untangle a web of related variables? Which variables are statistically significant and what role does each one play? Regression comes to the rescue because you can use it for all of these scenarios!

Regression in ML

- There is an important difference between classification and regression problems.
- Fundamentally, classification is about predicting a label and regression is about predicting a quantity.
- Predicting medical expenses using linear regression
  - In order for a health insurance company to make money, it needs to collect more in yearly premiums than it spends on medical care to its beneficiaries. As a result, insurers invest a great deal of time and money in developing models that accurately forecast medical expenses for the insured population.
- Predict cholesterol levels using data from EHRs

Simple Linear Regression

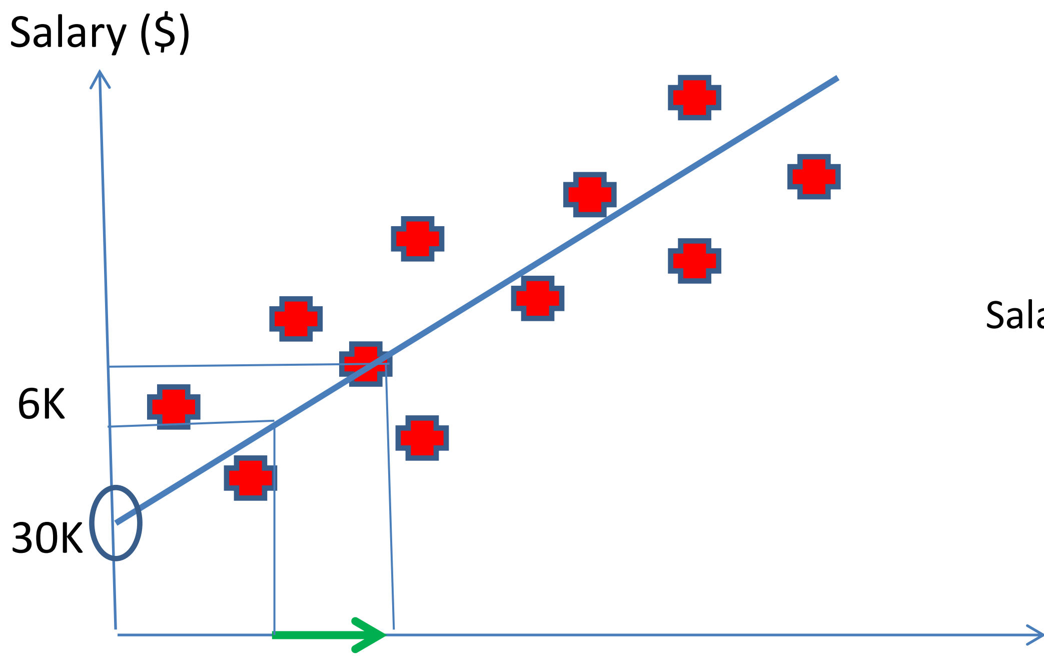
Constant

Coefficient

$$y = b_0 + b_1 * x_1$$

Dependent Variable (DV)

Independent Variable (IV)



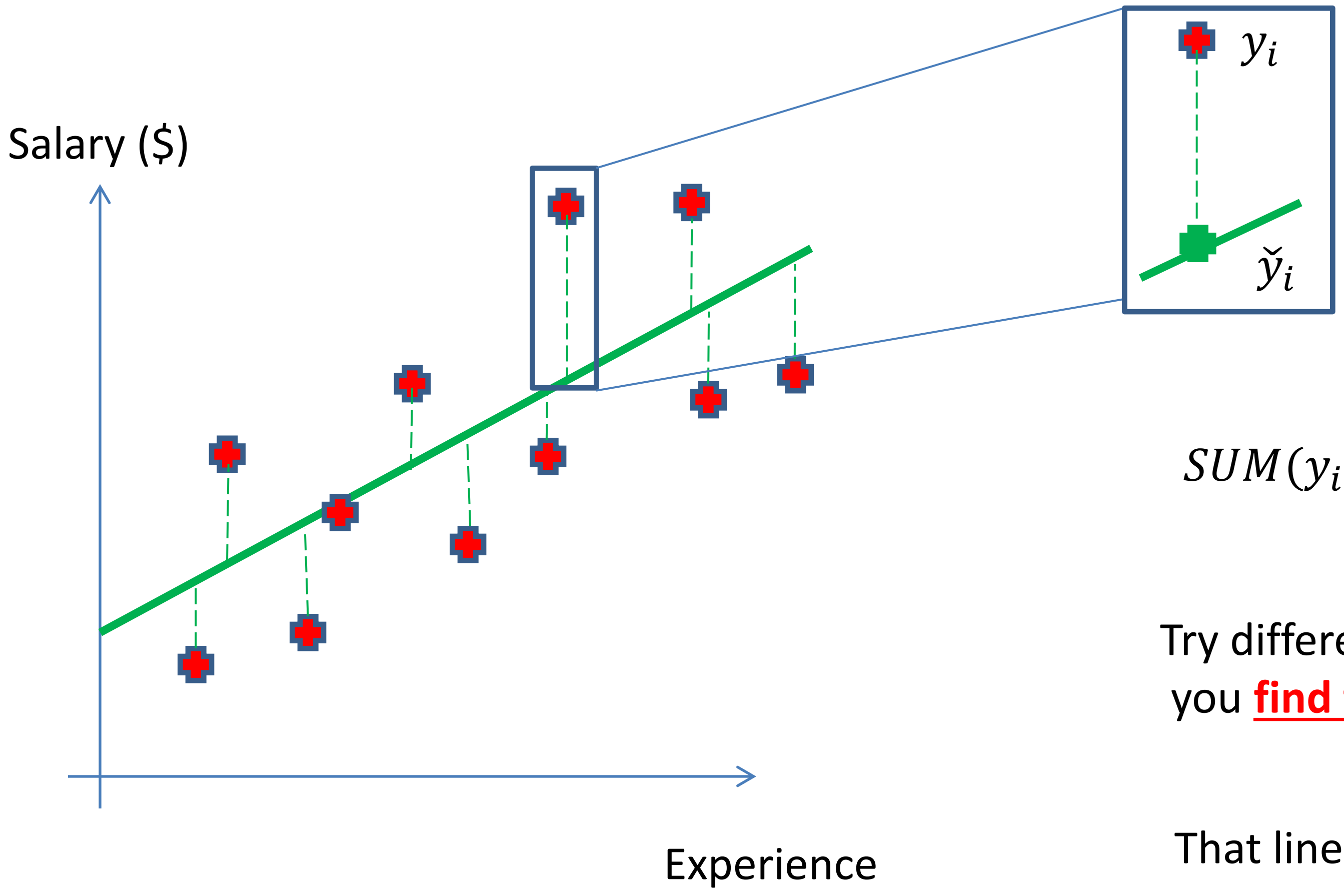
$$y = b_0 + b_1 * x_1$$

Salary = b_0 + b_1 * Experience

Slope of the line

Line that best
Fits this data

Ordinary Least Squares



$SUM(y_i - \check{y}_i)^2$

Try different lines until you **<u>find the min</u>**

That line is the best fit.

Multiple Linear Regression

Simple Linear Regression $y = b_0 + b_1 {*} x_1$

Multiple Linear Regression $y = b_0 + b_1 {*} x_1 + b_2 * x_2 + ... + b_n * x_n$

DV

Independent variables (IVs)

Caveats

- Assumptions of a Linear regression
    1. Linearity
    2. Homoscedasticity
    3. Multivariate normality
    4. Independence of errors
    5. Lack of multicollinearity

Lets look at the 50 startup company data

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	New York	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94

A VC firm wants to know if there is any correlation between profit and amounts spend
On R&D, administration, marketing and perhaps which state the startup is located.

Can we predict profit?

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	New York	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + + b_3 * x_3 + ??$$

We don’t have a number?
But we have nominal or
Categorical data

Approach: You need to create dummy variables

Dummy Variables

- First find out how many categories you have.
- Then add a new column with those names.
- Put a 1 or 0 depending upon if the company is located there

~~State~~	New York	Florida
~~New York~~	1	0
~~New York~~	1	0
~~Florida~~	0	1
~~New York~~	1	0
~~Florida~~	0	1

Dummy variables

Omit 1 dummy variable
Multicollinearity
$D_2 = 1 - D_1$

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + + b_3 * x_3 + b_4 * D_1$$

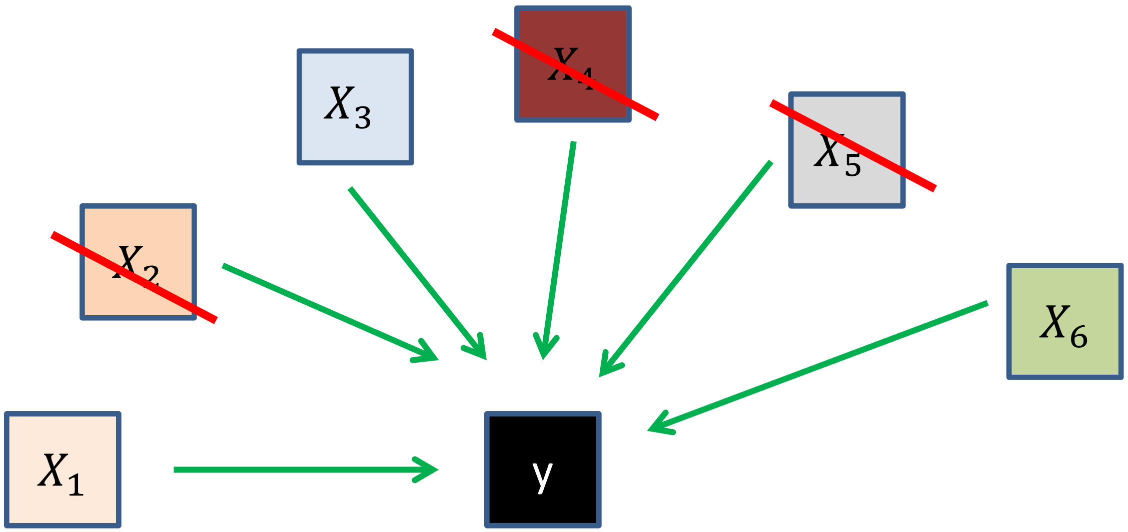
Repeat process for every categorical data

P-Value

- https://www.youtube.com/watch?v=KS6KEWaoOOE
- https://www.youtube.com/watch?time_continue=254&v=eyknGvncKLw

Multiple Linear Regression – Main Idea

- How to build a Model (Step-by-Step)



Why need to throw out?

1. GIGO
2. Explain these variables to management

5 methods to build a MLR model

1. All-in
2. Backward Elimination
3. Forward Selection
4. Bidirectional Elimination
5. Score Comparison

Stepwise Regression

All-in

- When?
- If you have prior knowledge; you know all these variables predict
- Mandatory – you have to use it based on company rules
- Preparing for Backward Elimination

Backward Elimination

- Step 1: Select a significance level to stay in the model (e.g., SL = 0.05)
- Step 2: Fit the full model with all possible predictors
- Step 3: Consider the predictor with the highest P-value.

  If P > SL, go to Step 4, otherwise go to FIN
- Step 4: Remove the predictor
- Step 5: Fit model without this variable

Keep removing until variable with highest P-value <= SL.

How Good are my Predictions?

MAE - Mean Absolute Error is the average of the difference between the Actual Values and the Predicted Values

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|$$

MSE - Mean Squared Error is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the **square** of the difference between the original values and the predicted values.

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

RMSE – Root Mean Square Error is the most popular evaluation metric used in regression problems. It follows an assumption that error are unbiased and follow a normal distribution

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

Train & Test Sets

	Area (sq m)	Bedrooms	Bathrooms	Price	
	200	3	2	$500,000	
	190	2	1	$450,000	y_train
X_Train	230	3	3	$650,000	
	180	1	1	$400,000	y_test
X_test	210	2	2	$550,000	

Scikit-learn provides us tools to train-test split and also advanced tools Called CV (Cross-fold Validation)

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, te
st_size = 0.2, random_state = 0)
```

Summary

- Regression models (both linear and non-linear) are used for predicting a real value, like salary for example. If your independent variable is time, then you are forecasting future values, otherwise your model is predicting present but unknown values.
- Multiple linear regression model works with numeric predictors and numeric label feature. It falls under supervised machine learning.
- It is among the top three most used method by data scientists and practitioners (others being decision tree and clustering).

Regularization

Introduction

- **Overfitting** is a phenomenon that occurs when a machine learning or statistics model is tailored to a particular dataset and is unable to generalize to other datasets. This usually happens in deep neural networks or even multiple regressions models.
- In order to create less complex (parsimonious) model when you have a large number of features in your dataset, Regularization techniques are used to address over-fitting and feature selections.

3 Types

- L1 Regularization
  - Lasso Regression
- L2 Regularization
  - Ridge regression
- Combining L1 and L2
  - Elastic Net
- A regression model that uses L1 regularization technique is called ***Lasso Regression*** and model which uses L2 is called ***Ridge Regression***.
  - *The key difference between these two is the penalty term.*

$$\hat{y} = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots\ldots + b_n * x_n$$

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Lasso Regression (Least Absolute Shrinkage and Selection Operator) or **L1 Regularization** adds "*absolute value of magnitude*" of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \text{RSS} + \boxed{\lambda \sum_{j=1}^{p} |\beta_j|}$$

λ is a hyperparameter we can tune

Ridge regression or L2 regularization adds "*squared magnitude*" of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \boxed{\lambda \sum_{j=1}^{p} \beta_j^2}$$

Elastic Net

- Elastic Net combines L1 and L2 with the addition of a alpha parameter deciding the ratio between them

$$\frac{\sum_{i=1}^{n}(y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

References & Sources

- *Introduction* to *Machine Learning* with *Python*: A *Guide* for *Data Scientists*. *Andreas C. Müller*, *Sarah Guido*. Publisher : O'Reilly Media; 1st edition (October 25, 2016) ISBN-13: 978-1449369415 ;ISBN-10: 1449369413
- Brandon Foltz, Statistics 101: Linear Regression, Algebra, Equations, and Patterns at https://www.youtube.com/watch?v=iAgYLRy7e20
- Kirill Eremenko & Hadelin de Ponteves, Super Data Science Workshop at Open Data Science Conference (ODSC) 2017.