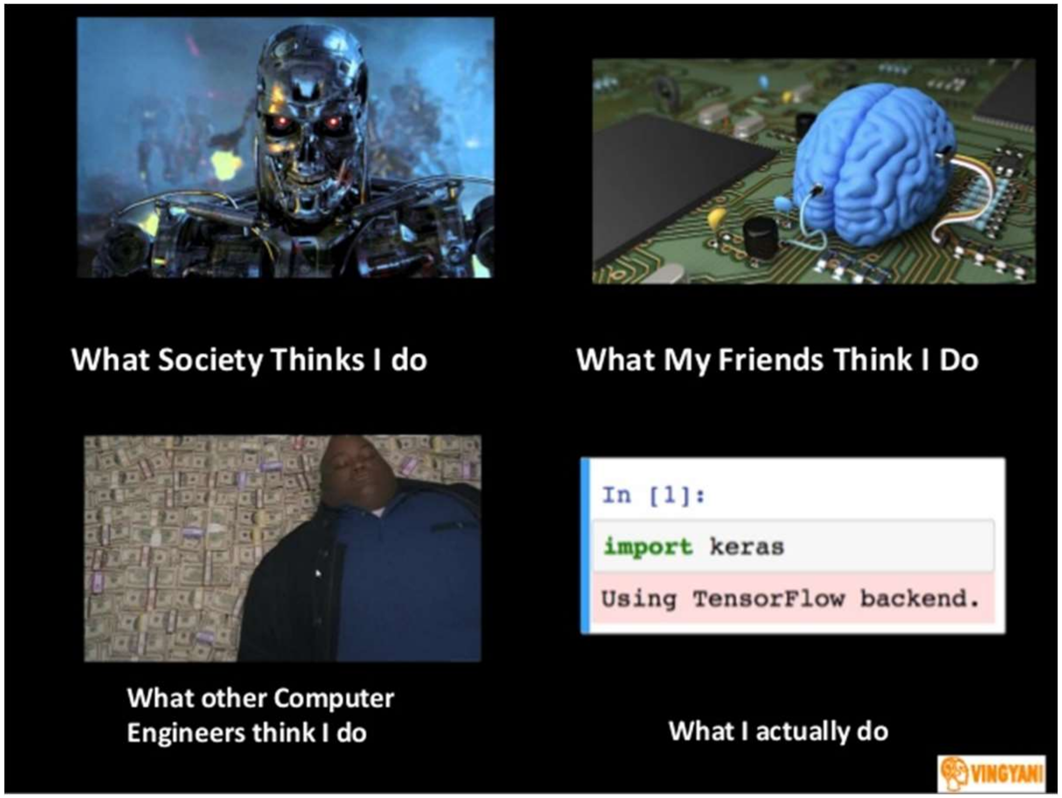
DATA SCIENCE 102:
INTRODUCTION TO MACHINE LEARNING

AGENDA

- What is Machine Learning?
- Types of Machine Learning
    - Applications of Machine Learning
- General Machine Learning Steps using Sci-Kit Learn
    - Fundamentals
        - Standard steps
        - Reading the library
    - Examples
        - Regression

WHAT IS MACHINE LEARNING

WHAT IS MACHINE LEARNING


What Society Thinks I do
What My Friends Think I Do
What other Computer Engineers think I do
What I actually do
VINGYANI

```
In [1]:
import keras
Using TensorFlow backend.
```

WHAT IS MACHINE LEARNING

A study of *algorithms and statistical models*

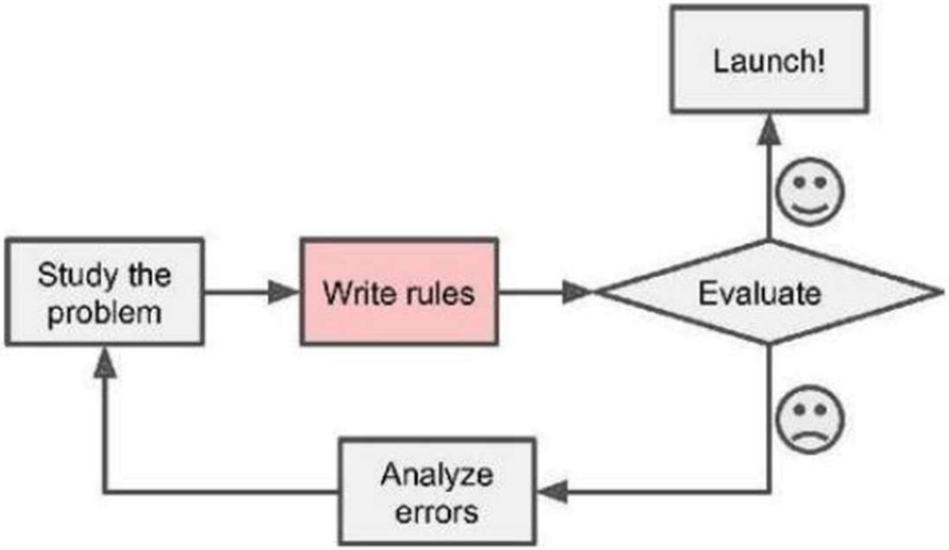
that computers use to perform a specific task effectively

without being explicitly programmed

EXAMPLE - SPAM EMAIL DETECTION

Without Machine Learning

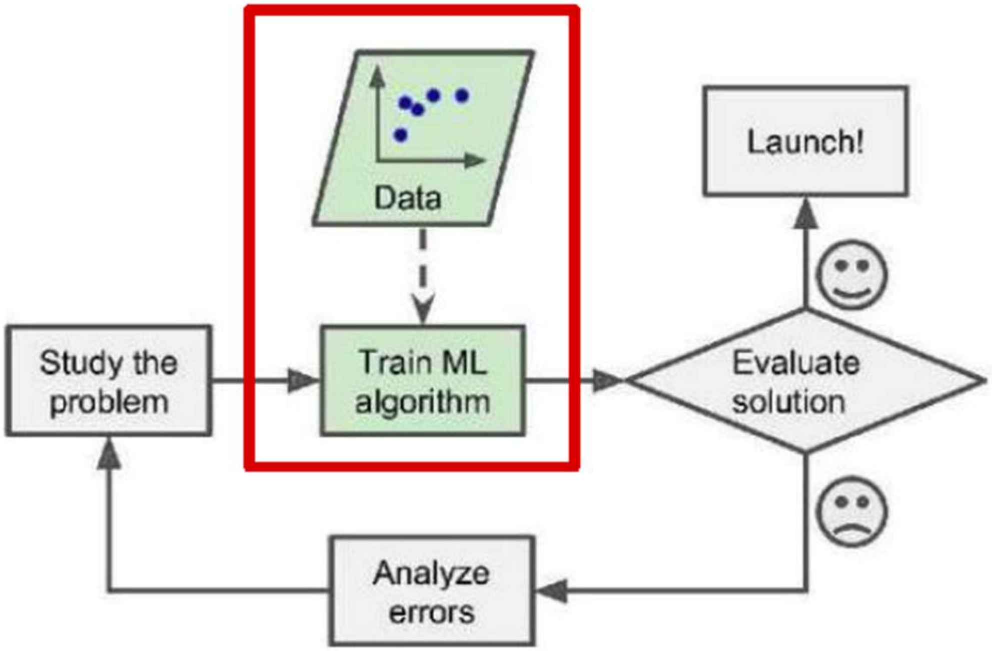
- You will write explicit rules to filter out emails containing words like:
    - *"4U", "Loan", "Prince", "Money Transfer", "Free"*
- Classify any email that contain these text as spam email



EXAMPLE - SPAM EMAIL DETECTION

With Machine Learning

- You train a model on some train data (spam & non-spam emails) with an algorithm that **infers** characteristics and patterns of spam emails
- The trained model can then apply learnt rules to detect signals of spam in an incoming email



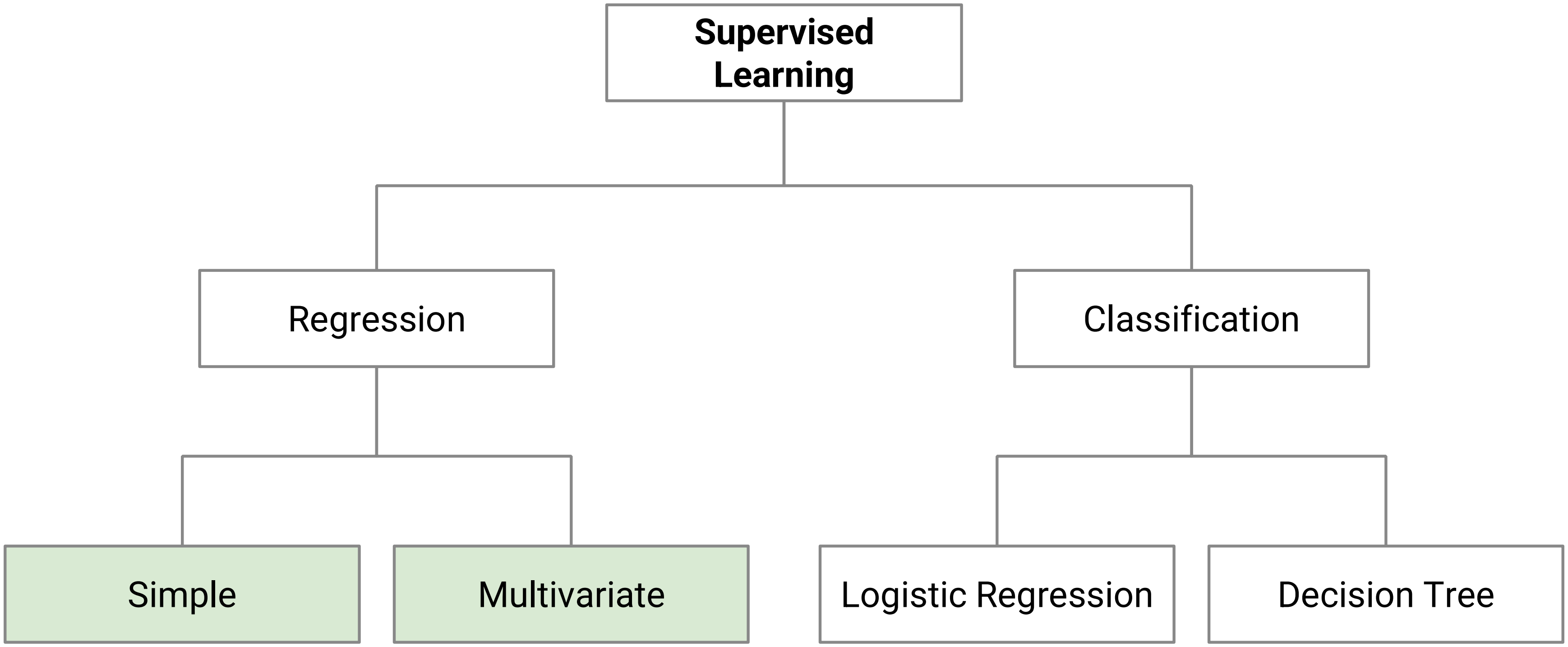
TYPES OF MACHINE LEARNING

- Types of Machine Learning
- Supervised Applications
- Unsupervised Applications

TYPES OF MACHINE LEARNING

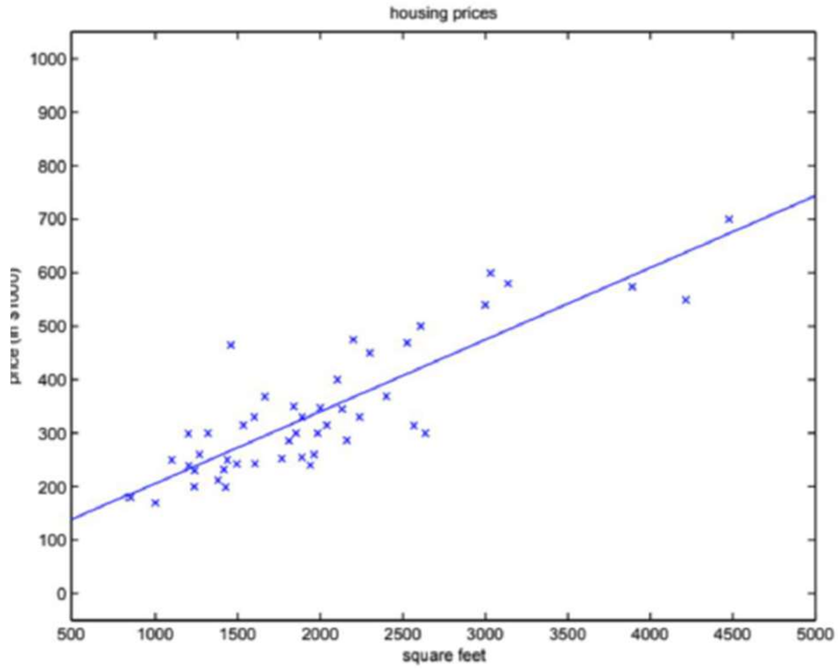
- Supervised Learning
  - Based on **labelled data,** makes **predictions** on a test set
- Unsupervised Learning
  - Based on **unlabelled data**, algorithm **discovers patterns** within the dataset

SUPERVISED LEARNING



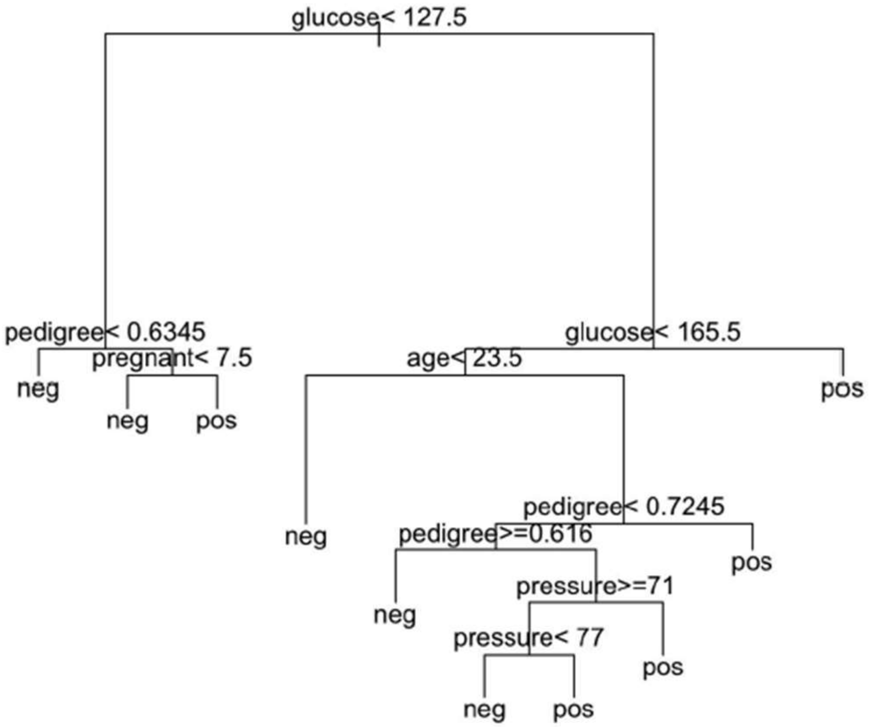
SUPERVISED APPLICATIONS

- Regression
  - Predict Housing Price (Target Variable) from Housing Floor Area (Input Feature)



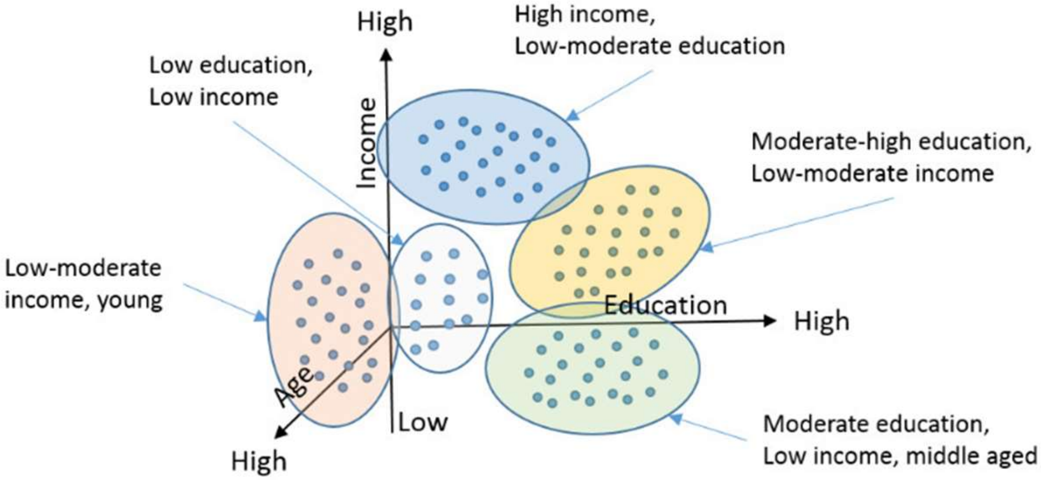
SUPERVISED APPLICATIONS

- Classification
  - Predict if a patient has diabetes (Yes/No)



UNSUPERVISED APPLICATIONS

- Clustering
    - Given a set of unlabelled data points, identify cluster each point belongs to



GENERAL MACHINE LEARNING STEPS USING SKLEARN

- Fundamentals
  - Standard Steps
  - Reading the library
- Examples
  - Regression

FUNDAMENTALS - STANDARD STEPS

When using any machine learning models with scikit learn, the following steps are usually applied in order:

Step 1: Choose a class of machine learning model from the library

Step 2: Choose the model’s hyperparameters by instantiating with desired values (tuning)

Step 3: Arrange data into features and target

Step 4: Fit model to your data by using the `fit()` method of the model

Step 5: Apply the model to new data:

- For supervised learning, using the `predict()` method
- For unsupervised learning, using the `predict()` or `transform()` method

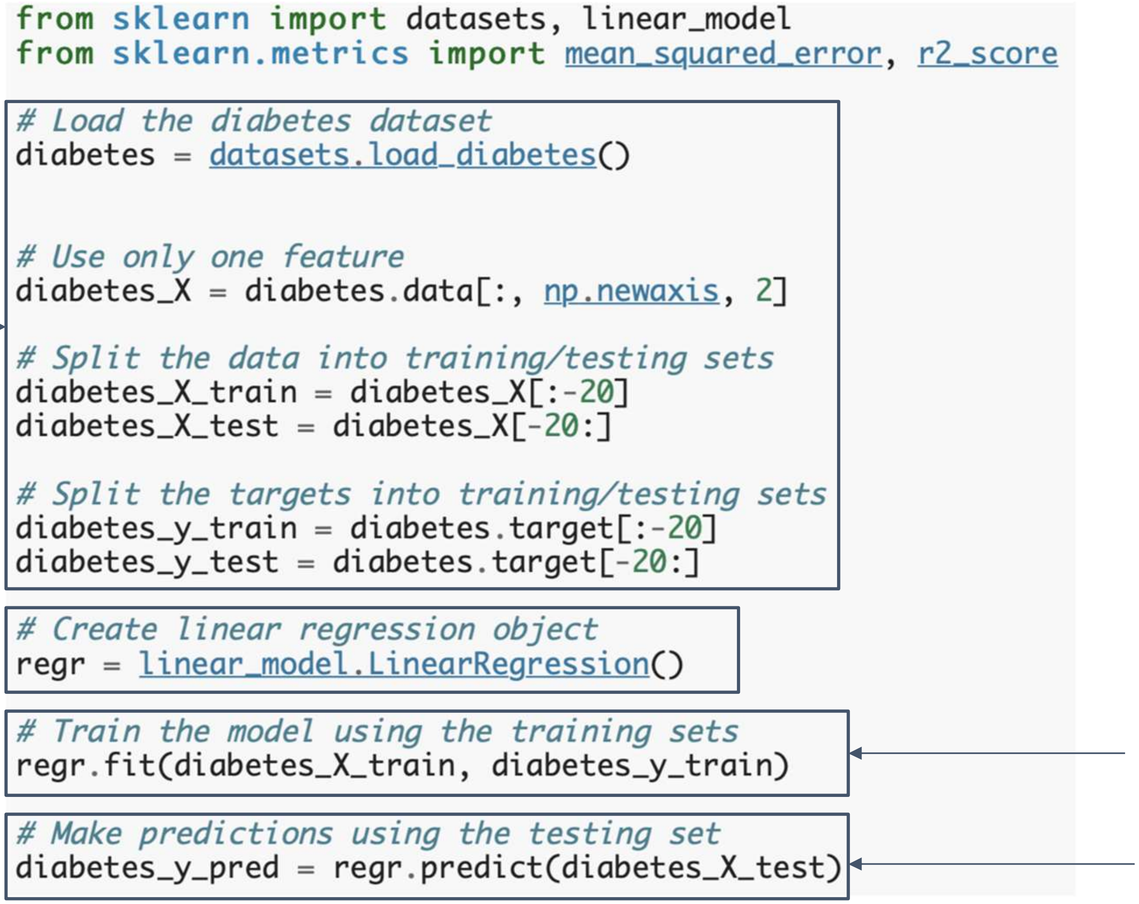
Python Data Science Handbook: Essential Tools for Working with Data by Jake VanderPlas (2016)

FUNDAMENTALS - READING THE LIBRARY

- Determine what model is used, then delve into details of that model in the library, and the other required parts of the library
- Here are some of the common classes you would use:
  - `sklearn.`<a name of a model> (i.e `sklearn.linear_model)`
    - Access to all functionalities of that model
  - `sklearn.metrics`
    - Functionalities of assessing the performance of your models
  - `sklearn.model_selection`
    - Performs cross validation and tuning of parameters
  - `sklearn.feature_selection`
    - Reduce dimension of dataset to boost performance

Python Data Science Handbook: Essential Tools for Working with Data by Jake VanderPlas (2016)

EXAMPLE - REGRESSION



```
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

Step 3

```
# Load the diabetes dataset
diabetes = datasets.load_diabetes()


# Use only one feature
diabetes_X = diabetes.data[:, np.newaxis, 2]

# Split the data into training/testing sets
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]

# Split the targets into training/testing sets
diabetes_y_train = diabetes.target[:-20]
diabetes_y_test = diabetes.target[-20:]
```

Step 1 & 2

```
# Create linear regression object
regr = linear_model.LinearRegression()
```

Step 4

```
# Train the model using the training sets
regr.fit(diabetes_X_train, diabetes_y_train)
```

Step 5

```
# Make predictions using the testing set
diabetes_y_pred = regr.predict(diabetes_X_test)
```

DATA SCIENCE 102: CLASSIFICATION

AGENDA

- What is Classification
- Logistic Regression
- Confusion Matrix

LEARNING OBJECTIVES

- Build an intuition for Logistic Regression
- Able to interpret results and performance from Logistic Regression
- Apply Confusion Matrix to evaluate classification model

LOGISTIC REGRESSION

- Intuition
- Confusion Matrix

Logistic Regression

- Logistic Regression is one of the basic and popular algorithms to solve a binary classification problems
- For each input, logistic regression outputs a probability that this input belongs to the 2 classes
  - Set a probability threshold boundary and that determines which class the input belongs to
- Binary classification problems (2 classes):
  - Emails (Spam / Not Spam)
  - Transactions (Fraudulent / Not Fraudulent)
  - Loan Default (Yes / No)

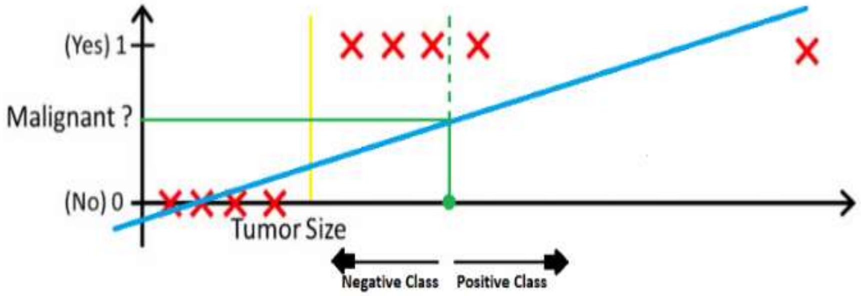
Why Not Use Linear Regression for Classification?

- Linear Regression is used to solve regression problems with continuous values
- Logistic Regression is used to solve classification problems with discrete categories
  - Binary classification (Classes 0 and 1)

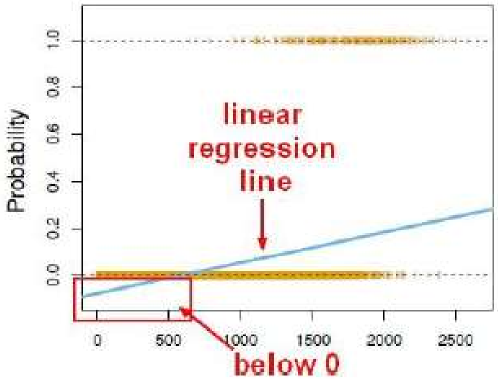
Why Not Use Linear Regression for Classification?

- Let’s say we want to predict the malignancy of a tumour based on its size
- We then plot a simple linear regression line and set the threshold as 0.5
  - Negative class – Tumour size on the left size
  - Positive class – Tumour size on the right size

Linear regression line is not a good fit

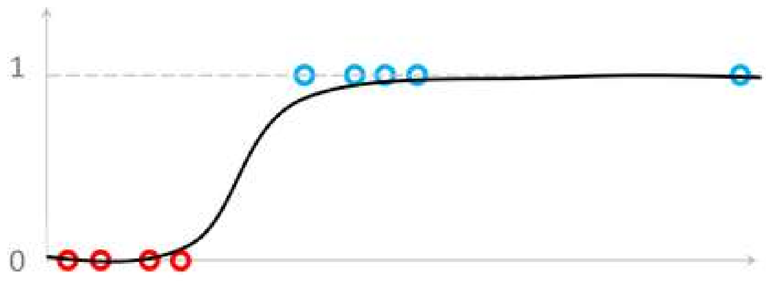


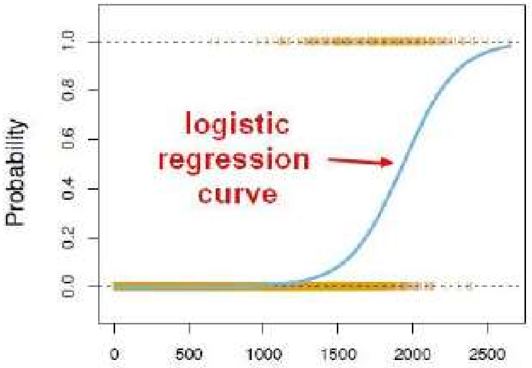
Another example: Negative probability



Why Not Use Linear Regression for Classification?

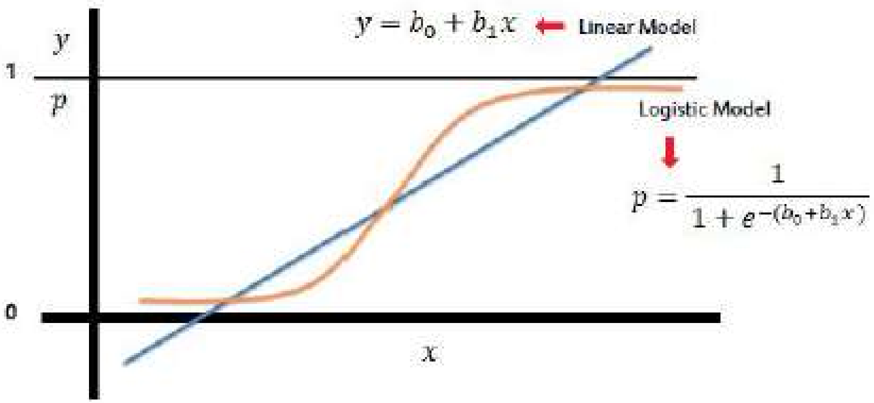
- Solution – Transform linear regression to a logistic regression curve
- Logistic regression is a Sigmoid function
  - Sigmoid function takes in any real value
  - Output probability between 0 and 1





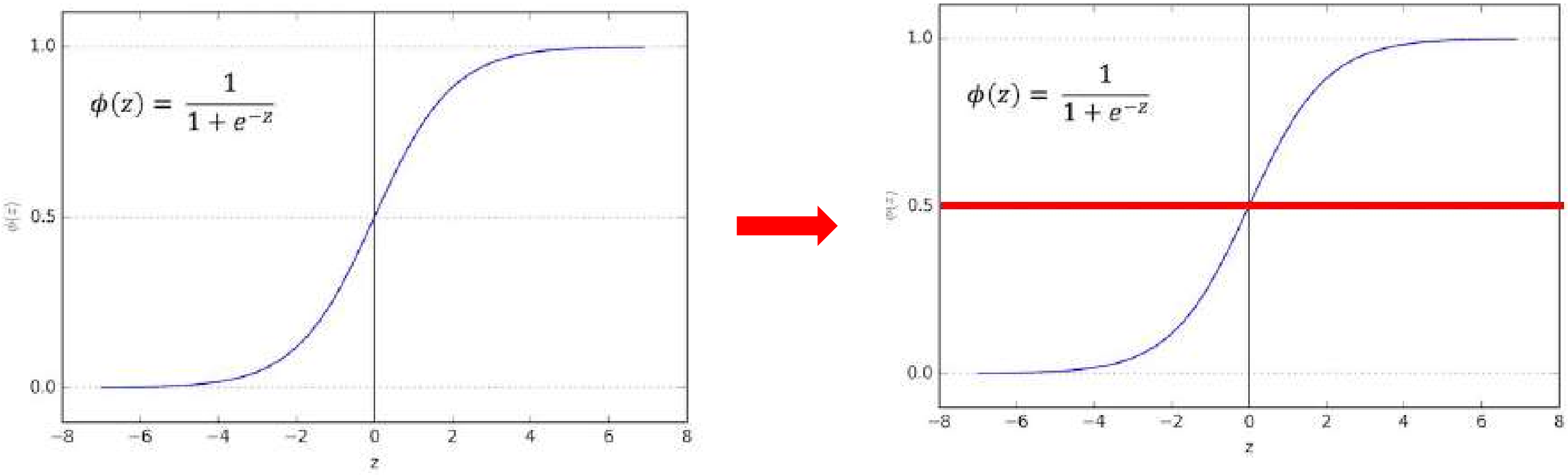
Sigmoid Function (Logistic Function)

- Take linear regression and put in into the Sigmoid function
- Sigmoid function outputs probability between 0 and 1



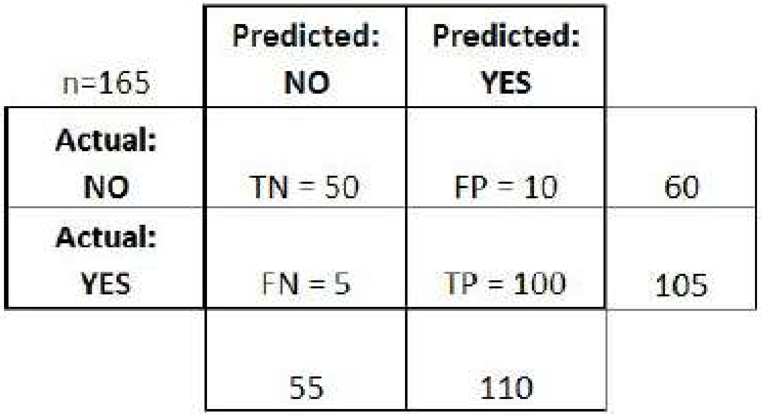
Sigmoid Function (Logistic Function)

- Sigmoid function outputs probability between 0 and 1
- Default probability threshold is set as 0.5 typically
    - Class 0 – Below 0.5
    - Class 1 – Above 0.5



Model Evaluation – Confusion Matrix

- To evaluate logistic regression model, confusion matrix is typically used for model evaluation
- Imagine now a doctor wants to use a logistic regression model to predict if patients have (or don't have) disease



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

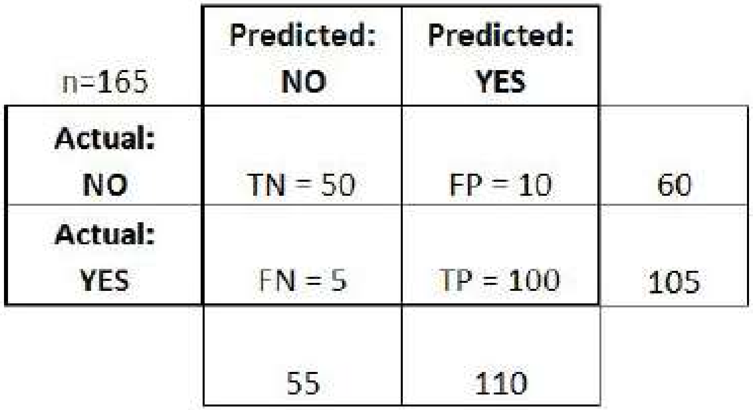
- True Positives (TP)
- True Negative (TN)
- False Positives (FP)
- False Negatives (FN)

Metrics for Model Evaluation is Classification

- Accuracy
  ➢ $\frac{TP+}{Total}$

- Precision
  ➢ $\frac{TP}{TP+}$

- Recall
  ➢ $\frac{TP}{TP+FN}$



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Precision: 100/110 → 90.9%

Recall: 100/105 → 95.2%

Example - Disease Detection

- We want to focus on Recall as the main metric
- Reduce FN and increases Recall
- Because if an infected patient goes through the test undetected, the risk is very high if the disease is contagious

DATA SCIENCE 102: REGRESSION

AGENDA

- What Is Regression
  - Applications of Regression
- Simple Linear Regression
  - Intuition
  - Interpretation
  - Ordinary Least Squares
  - Measure of Fit
  - Model Validation - Train Test Split
  - Performance Measures
  - Scikit Learn Example and Practice
- Multivariate Linear Regression
  - Interpretation
  - Feature Selection
  - Scikit Learn Example and Practice

LEARNING OBJECTIVES

- Build an intuition for Linear Regression
- Able to interpret results and performance from Linear Regression
- Discern difference between Single and Multivariate Linear Regression
- Importance of Feature Selection in Multivariate Linear Regression

WHAT IS REGRESSION

- Applications of regression

WHAT IS REGRESSION

- Regression Analysis is a model to develop an equation that shows how variables are related
- In regression terminology, the variable being predicted is the **dependent variable** and the variables being used to predict the dependent variable are called **independent variables** (predictor variables)
- Regression can be used in the following cases:
  - Predicting continuous labels
  - Classification (logistic regression)

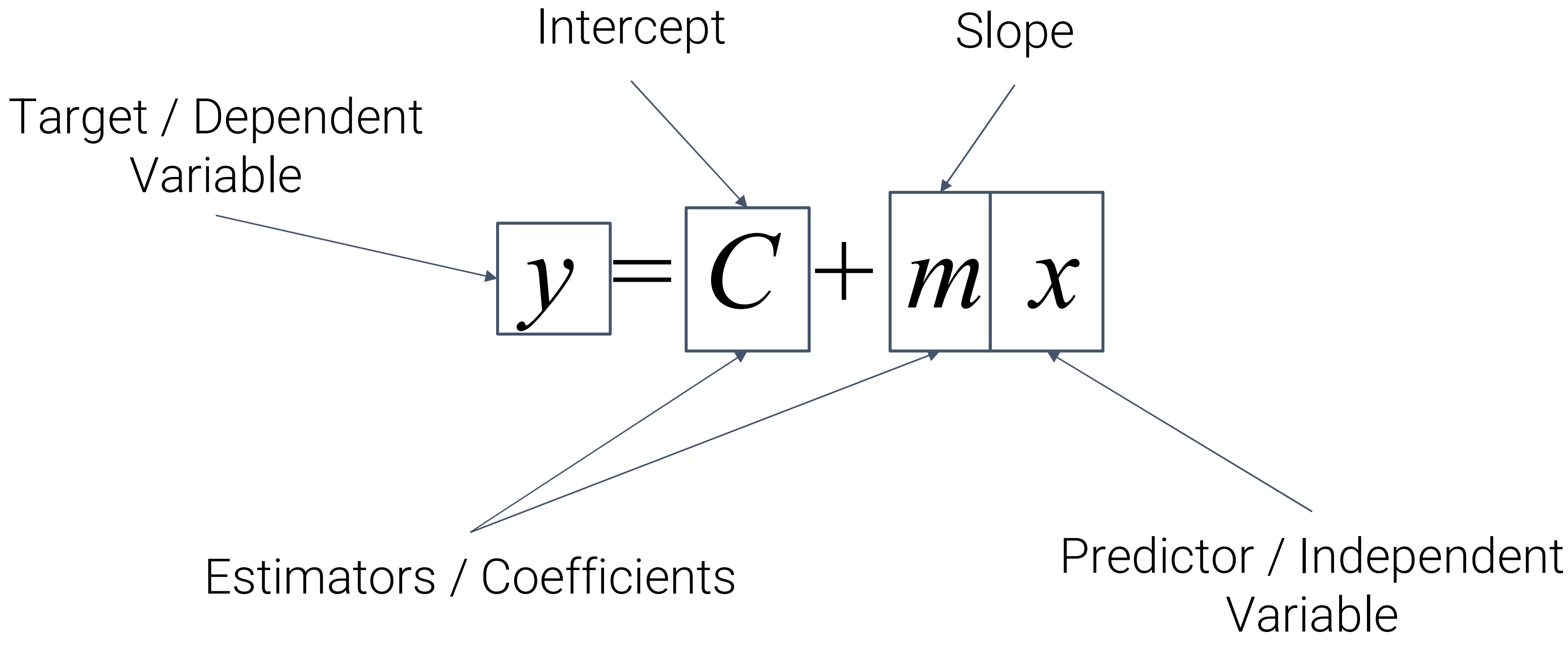
An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

SIMPLE LINEAR REGRESSION

- Intuition
- Single vs Double variable
- Interpretation
- Ordinary Least Squares
- R Squared and Adjusted R Squared
- Performance Measures
- Train-Test-Split

INTUITION - THE REGRESSION EQUATION

Remember the y = mx + c you learnt in secondary school? That is essentially regression!



Intercept

Slope

Target / Dependent Variable

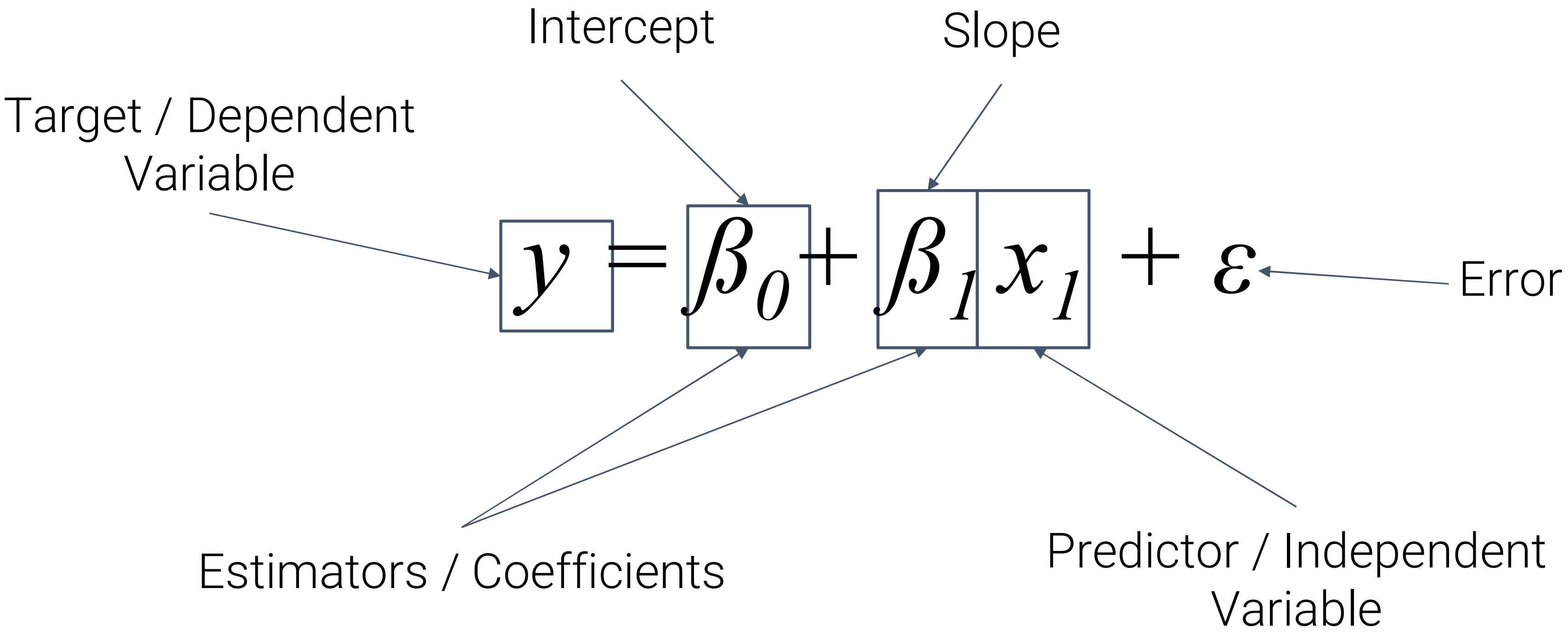
$$\boxed{y} = \boxed{C} + \boxed{m}\boxed{x}$$

Estimators / Coefficients

Predictor / Independent Variable

INTUITION - THE REGRESSION EQUATION

This is the adult, and machine-learning way of representing regression:



$$\boxed{y} = \boxed{\beta_0} + \boxed{\beta_1}\boxed{x_1} + \varepsilon$$

Intercept

Slope

Target / Dependent Variable

Error

Estimators / Coefficients

Predictor / Independent Variable

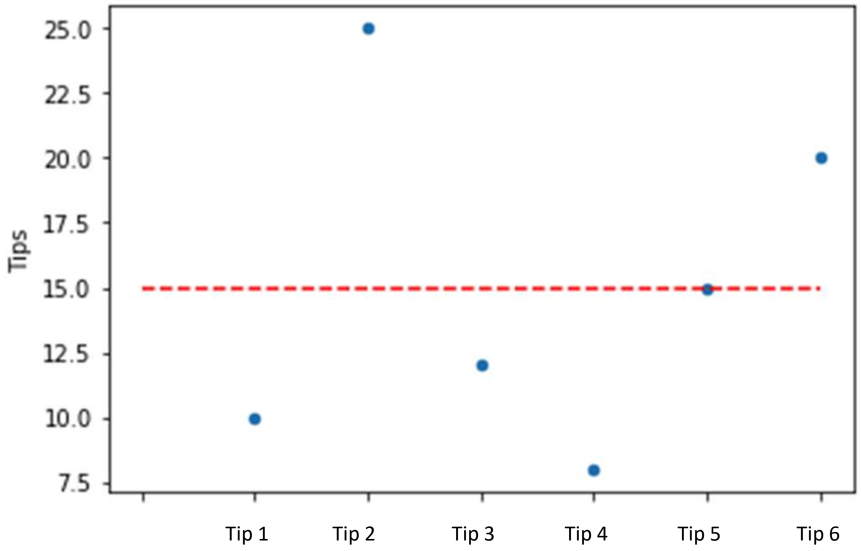
STUDY OF SINGLE VARIABLE - NO VALUE ON ITS OWN

- Predict future tips given the following dataset:
    - To “predict” future tips, use <u>average</u>

$$Tip = 0\ (No\ Predictor) + 15$$

$$Tip = 15$$

Tip #	Amount ($)
Tip 1	10
Tip 2	25
Tip 3	12
Tip 4	8
Tip 5	15
Tip 6	20

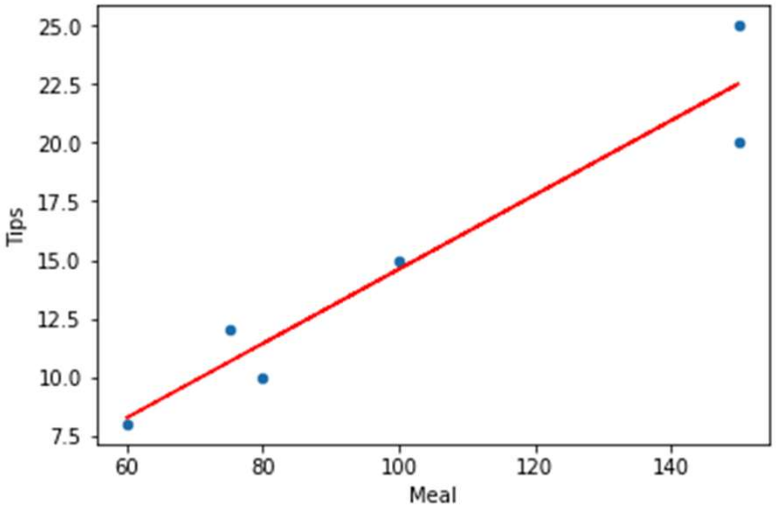


STUDY OF TWO VARIABLES - SIMPLE LINEAR REGRESSION

- Predict future tips given the following dataset
  - To “predict” future tips using linear regression

Tip #	Amount ($)	Meals ($)
Tip 1	10	80
Tip 2	25	150
Tip 3	12	75
Tip 4	8	60
Tip 5	15	100
Tip 6	20	150

$$Tip = -1.27 + 0.158\ (Meal)$$



INTERPRETATION

- After fitting the model, you can extract the coefficients (estimators) of the model by using `<variablenameofmodel>.coeff_`. To get the intercept, use, `<variablenameofmodel>.intercept_`
- Let's say given the following regression equation:

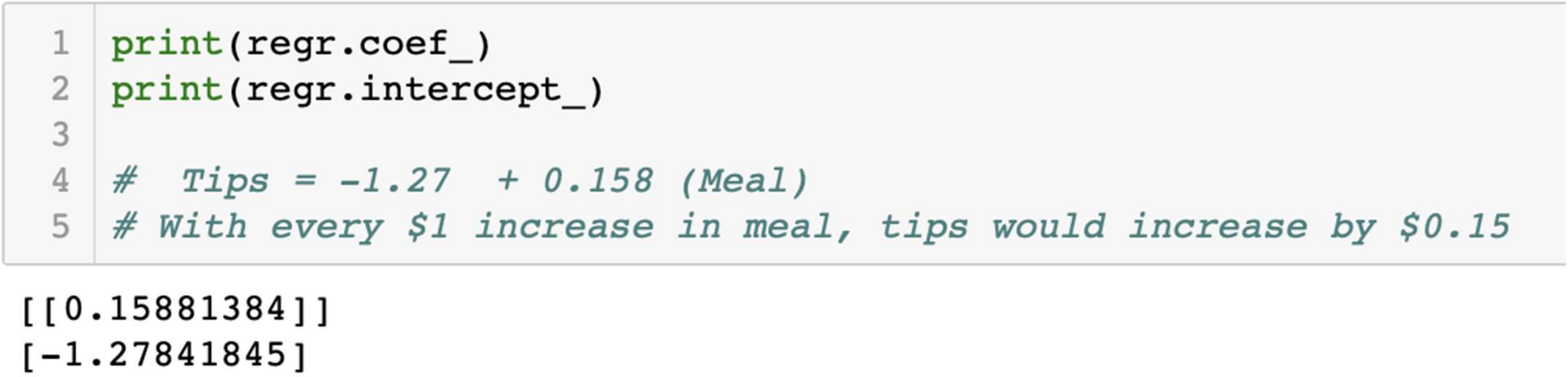
$$Tip = -1.27 + 0.158\ (Meal)$$

- We can interpret it as, given ***1 dollar increase*** in the cost of a meal, there will be an **increase** of tips by ***0.158 dollars***

INTERPRETATION - CODED EXAMPLE

- Let’s say given the following regression equation:

$$Tip = -1.27 + 0.158\ (Meal)$$



```python
print(regr.coef_)
print(regr.intercept_)

#  Tips = -1.27  + 0.158 (Meal)
# With every $1 increase in meal, tips would increase by $0.15
```

```
[[0.15881384]]
[-1.27841845]
```

- So what happens behind `.fit()` == OLS

HOW LINEAR REGRESSION EQN IS CONSTRUCTED - OLS

- Now we understand Linear regression is about creating an equation that shows how variables are related. In code form your “`.fit()`” will create this equation.

- This equation is useful in helping us (1) interpret relationship, and (2) predict output based on unseen input data

Means we can sub in any X-Value (meal) to predict tips

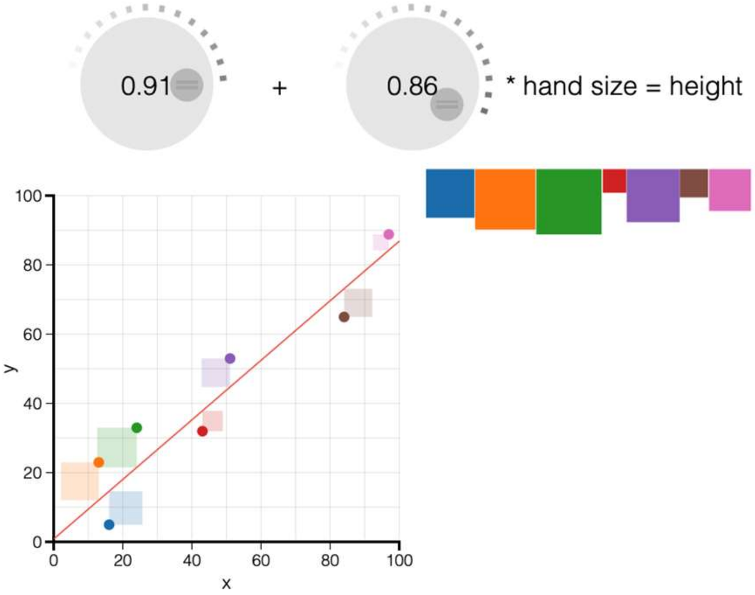
$$Tip = -1.27 + 0.158\,(Meal)$$

- Then the question is, how do we derive the y-intercept (β_0) and gradient (β_1)? Answer is: Ordinary Least Squares (OLS).

ORDINARY LEAST SQUARES - VISUALISATION OF PROCESS

- Ordinary Least Squares (Least Squares Method) is aimed at **minimizing** the sum of squared residuals (SSR), i.e keep the errors (ε) as low as possible

Annotation on betas



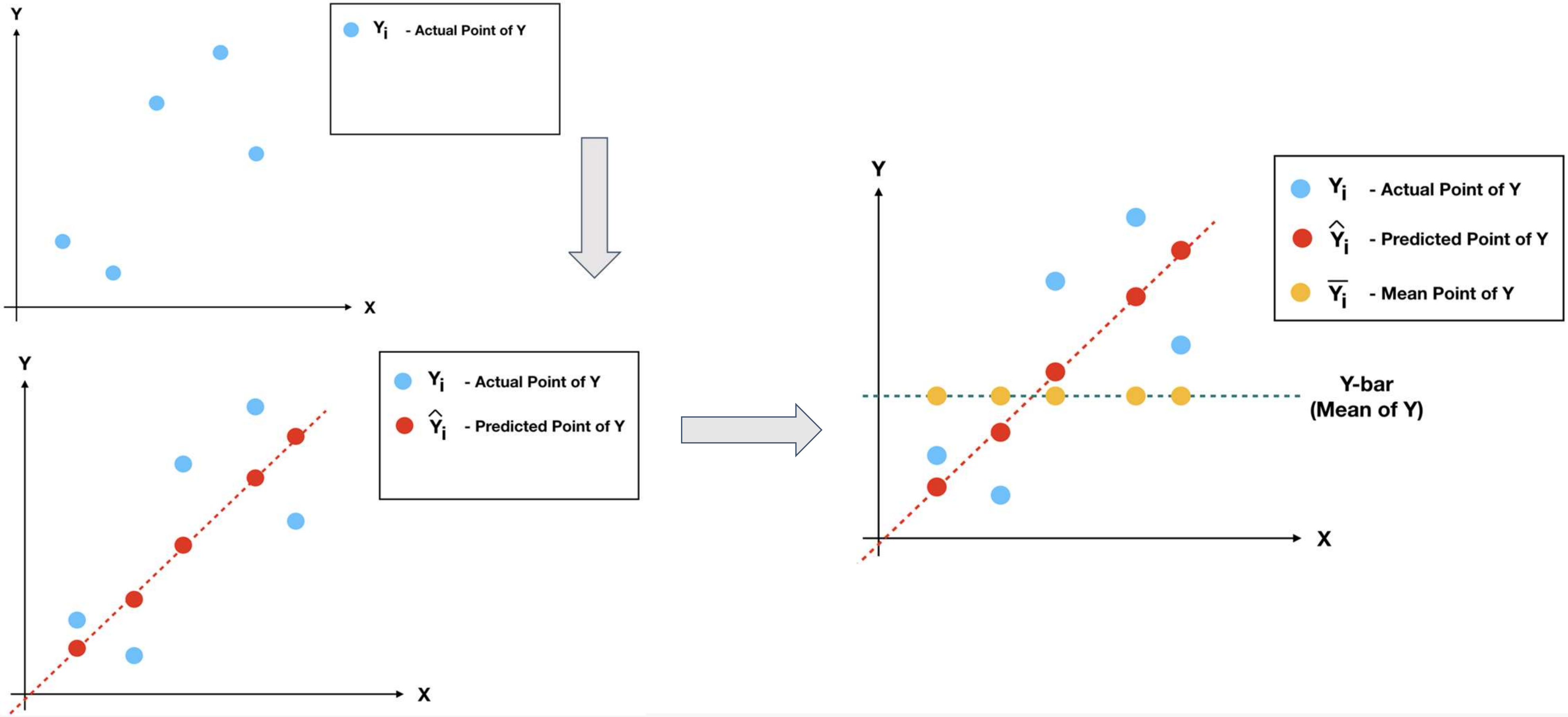
Click here to view a visualisation of OLS in action: http://setosa.io/ev/ordinary-least-squares-regression/

ORDINARY LEAST SQUARES - KEY CONCEPT

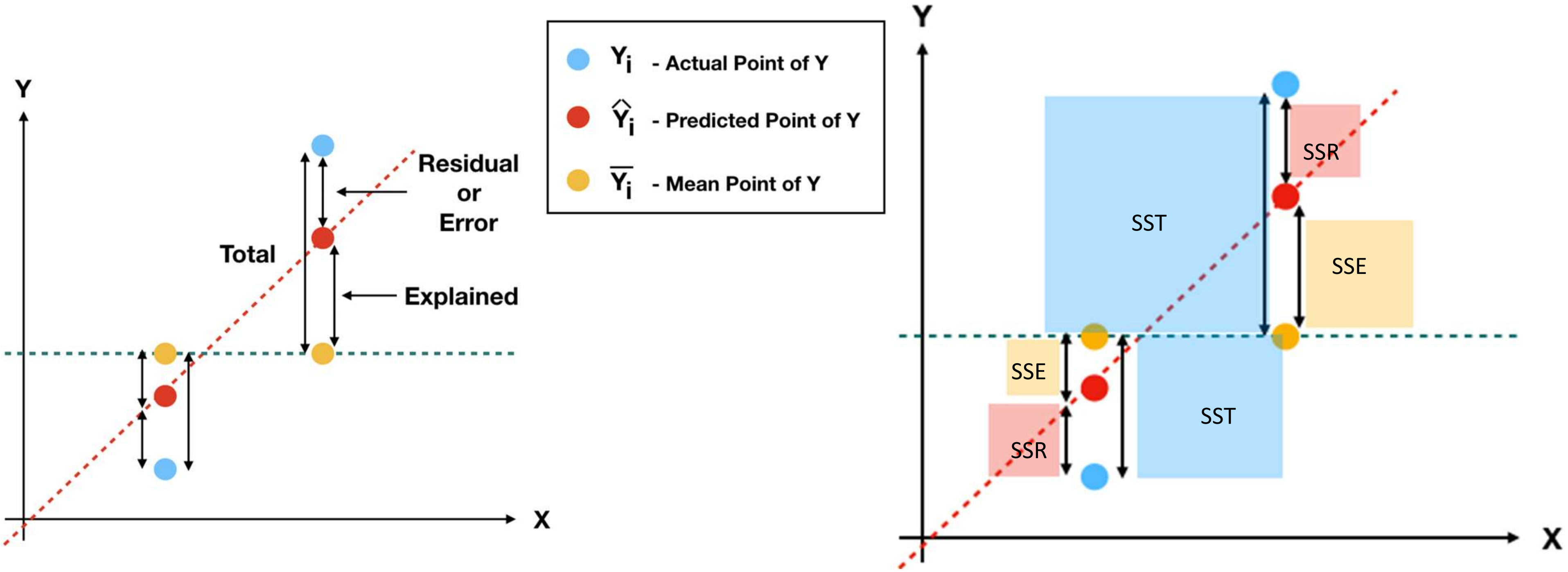
- Intuitively, OLS is fitting a line through the sample points such that the sum of squared errors (SSR) is as small as possible, hence the term least squares
- The OLS regression line always goes through the mean of the sample (y)
- The residual, e, is an estimate of the error term, ε, and is the difference between the fitted line (sample regression function) and the sample point
- The sum of the OLS residuals is zero

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

ORDINARY LEAST SQUARES - BUILDING BLOCKS EXPLAINED


Y
X
● Y_i - Actual Point of Y
Y
X
● Y_i - Actual Point of Y
● $\hat{Y}_i$ - Predicted Point of Y
Y
X
● Y_i - Actual Point of Y
● $\hat{Y}_i$ - Predicted Point of Y
● $\overline{Y}_i$ - Mean Point of Y
Y-bar
(Mean of Y)

ORDINARY LEAST SQUARES - BUILDING BLOCKS EXPLAINED


Y
X
Residual
or
Error
Total
Explained
● Y_i - Actual Point of Y
● $\hat{Y}_i$ - Predicted Point of Y
● $\overline{Y}_i$ - Mean Point of Y
Y
X
SST
SSR
SSE
SSE
SSR
SST

MEASURING FIT BETWEEN VALUES AND PREDICTED LINE - R^2

- A measure of how well the sample values fit the predicted regression line is **goodness of fit**

- R (also known as Pearson's R) is the sample correlation coefficient; ranges from -1 to 1; >0 = positive correlation, <0 = negative correlation

- R^2 (R-squared) is known as the coefficient of determination; range from 0 to 1

$$\frac{\text{SSE}}{\text{SSR} + \text{SSE}} = \frac{\text{SSE}}{\text{SST}} = \text{R}^2$$

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

CALCULATION OF R^2

- Each observation (y_i) on the linear regression as made up of an explained part ($\hat{y}_i$) and an unexplained part (e_i):

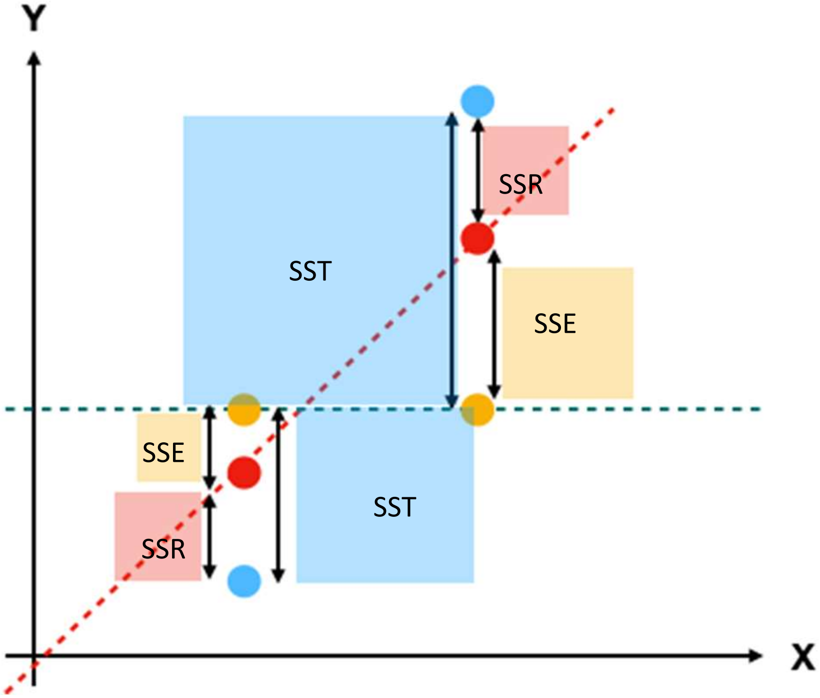
$$y_i = \hat{y}_i + e_i$$

Total sum of squares (SST) = $\sum (y_i - \bar{y}_i)^2$

Explained sum of squares (SSE) = $\sum (\hat{y}_i - \bar{y}_i)^2$

Residual sum of squares (SSR) = $\sum (y_i - \hat{y}_i)^2$

SST = SSE + SSR



An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

AN ALTERNATIVE OF MEASURING FIT - ADJUSTED R^2

- Adjusted R^2 modifies the original R^2 by incorporating the sample size and the number of explanatory variables in the model

- Can be found in `sklearn.metrics.r2_score` or `LinearRegression.score()`

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

R-SQUARED VS. ADJUSTED R-SQUARED

- Both R^2 and the adjusted R^2 (Adj-R^2) gives an idea of how many data points fall within the line of the regression equation
- The main difference is:
  - R^2 tells you every single variable explains the *variation* in the depend variable (y)
  - Adj-R^2 tells you the **percentage of variation** explained only by the independent variables that actually affect the dependent variable

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

R-SQUARED VS. ADJUSTED R-SQUARED

- Formula for Adj-R^2 :

$$\text{Adjusted } R^2 \ (R^2_{adj}) = 1 - \left[\frac{(1-R^2)(n-1)}{n-k-1}\right]$$

  - n is the number of points in your data sample
  - k is the number of independent variables (predictors) excluding constants

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

R-SQUARED VS. ADJUSTED R-SQUARED

- The adjusted R^2 (Adj-R^2) penalizes for adding more independent variables that do not fit the model.
- As R^2 increases with every predictor added to a model, it can appear to be a **better fit** with more terms added to the model, but this can be misleading
- Adding too many variables and polynomials may run into the trouble of **overfitting**, thus a misleading high R^2 value can lead to misleading projections
- In short, adjusted R^2 is preferred over R^2 in multivariate linear regression.

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

MEASURING FIT VS MEASURING PERFORMANCE

- Measuring Fit - Quantified via R^2 / Adjusted $R^{2,}$ is a measure of how well the seen / known values fit the predicted regression line

- Measuring Performance - is a measure of how well the predicted regression line can predict unseen/unknown values (Quantified RMSE or MAE)

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

PERFORMANCE MEASURES - LINEAR REGRESSION

- For linear regression there are several measures to assess the performance of the model:
  - **MAE** (`sklearn.metrics.mean_absolute_error`)
    - Mean Absolute Error
    - Whether predictions are, on average, over/under predicting the outcome
  - **RMSE** - (`sklearn.metrics.mean_squared_error` then use ** 0.5)
    - Root Mean Squared Error
    - Differences between predicted vs observed/actual values
    - Similar to standard error
    - Lower is better
    - 0 means perfect fit to data *(not the best, could be overfitting)*
- Can be found in the `sklearn.metrics` part of `sklearn`

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

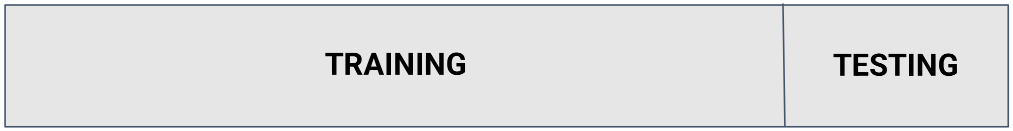
PERFORMANCE MEASURES - LINEAR REGRESSION

- How are RMSE and MAE derived?
- Answer: Via Model-Validation (Train-Test-Split)

An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

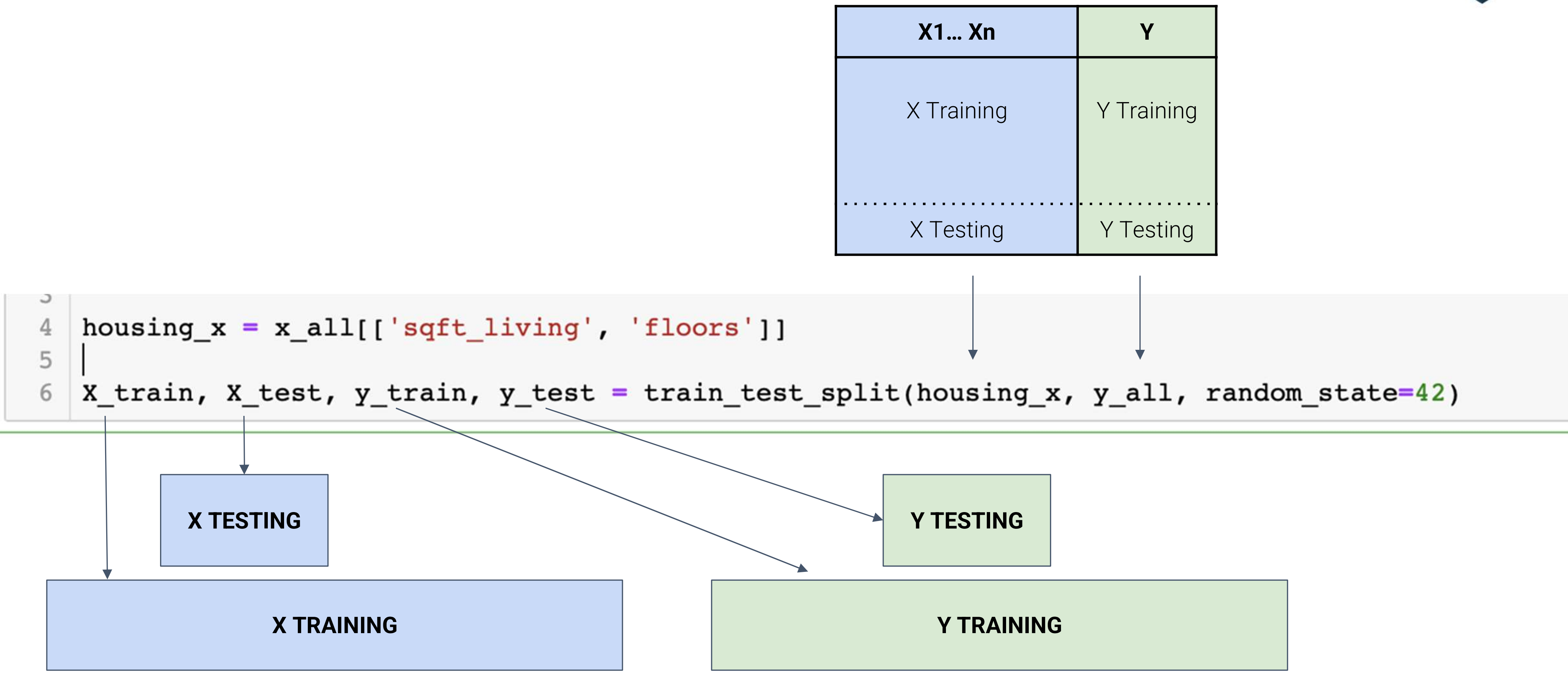
MODEL VALIDATION - TRAIN TEST SPLIT

- *After selecting and fitting* a machine learning model, like Linear Regression for example, we need to ensure that we validate our model **by comparing some of the training data and comparing the prediction against its known value**
- One of the ways to validate the model would be to use *Holdout Sets*, basically splitting the given datasets for training and testing
- We can use Scikit learn's `train_test_split` to split our datasets
- By default, it is an 80-20 split (80% training, 20% testing)



An Introduction to Statistical Learning (with applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (2013)

MODEL VALIDATION - TRAIN TEST SPLIT - CODED EXAMPLE



X1… Xn	Y
X Training	Y Training
X Testing	Y Testing

```
housing_x = x_all[['sqft_living', 'floors']]

X_train, X_test, y_train, y_test = train_test_split(housing_x, y_all, random_state=42)
```

SCIKIT LEARN EXAMPLE AND PRACTICE*

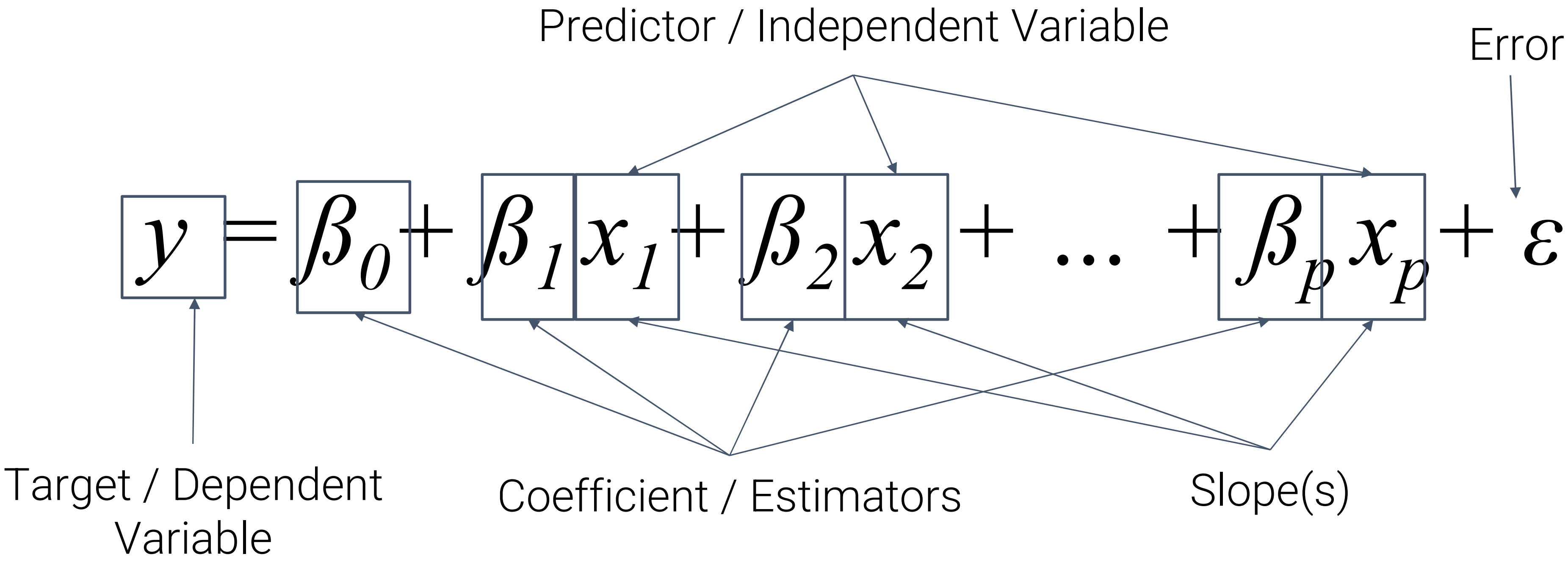
- Try out the practice in your in-class notebook 5
- Remember the 5 common steps for using SKLearn

MULTIVARIATE LINEAR REGRESSION

- Intuition and Interpretation
- R-Squared vs. Adjusted R-Squared
- Feature Selection

INTUITION

- Multivariate linear regression incorporates **more than one predictors** into the equation



Predictor / Independent Variable

Error

$$\boxed{y} = \boxed{\beta_0} + \boxed{\beta_1}\boxed{x_1} + \boxed{\beta_2}\boxed{x_2} + \ldots + \boxed{\beta_p}\boxed{x_p} + \varepsilon$$

Target / Dependent Variable

Coefficient / Estimators

Slope(s)

INTERPRETATION

- Let’s say given the following regression equation:

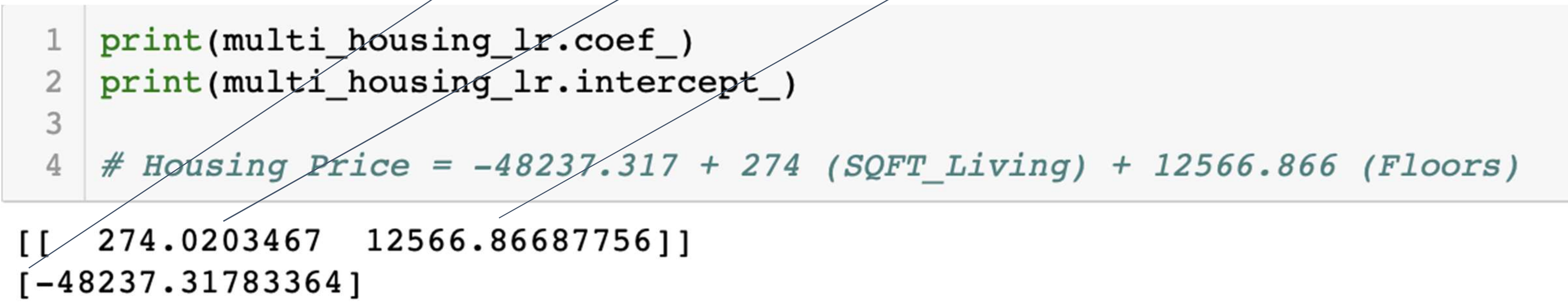
$$HousingPrice\ (\$) = -\ 48237.317 + 274\ SQFT + 12566.866\ Floor + \varepsilon$$

- We can interpret it as, given 1 SQFT increase, Housing Price increases by $274; given 1 floor increase, Housing Price increase by $12566.866

INTERPRETATION - CODED EXAMPLE

- Let’s say given the following regression equation:

$$\textit{HousingPrice}\ (\$) = -\ 48237.317 + 274\ \textit{SQFT} + 12566.866\ \textit{Floor} + \varepsilon$$



```
print(multi_housing_lr.coef_)
print(multi_housing_lr.intercept_)

# Housing Price = -48237.317 + 274 (SQFT_Living) + 12566.866 (Floors)
```

```
[[  274.0203467   12566.86687756]]
[-48237.31783364]
```

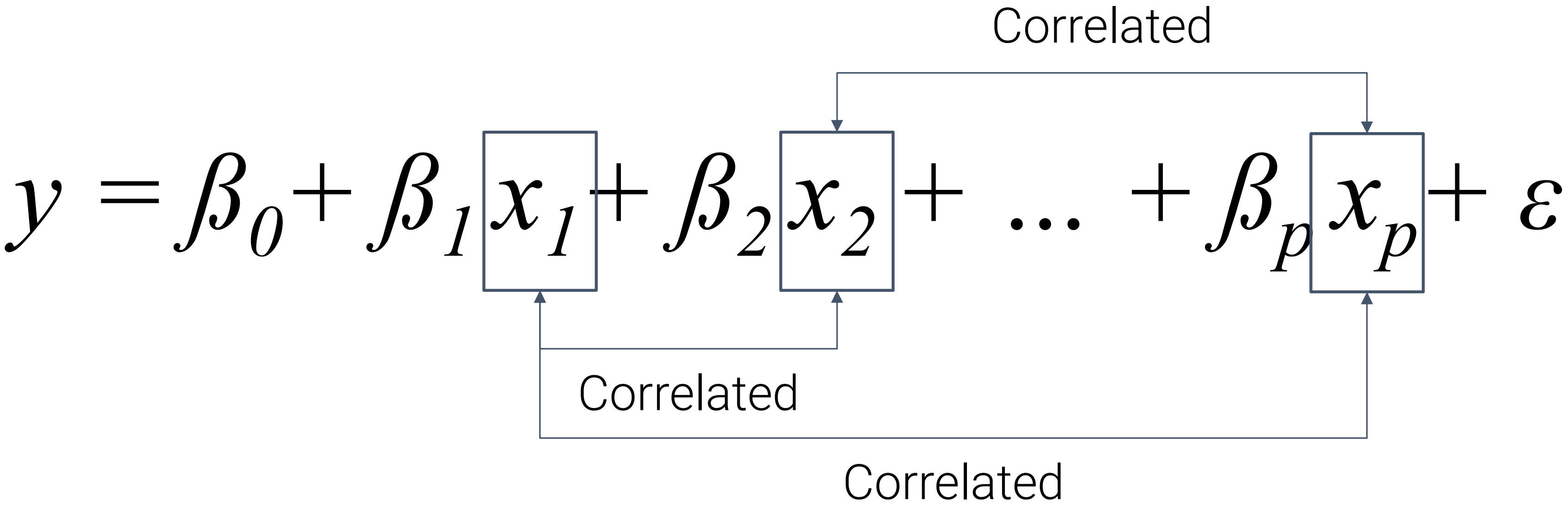
FEATURE SELECTION

- Why don’t we just use all the variables in the world and just apply it to our model?
  - Expensive or not feasible
  - Sometimes fewer predictors are better
  - More predictors could lead to possibly more missing data
  - Lesser predictors allow for greater insight into “influence”
  - Unstable regression coefficient due to multicollinearity
- Approach to reducing / selecting predictors:
  - Domain expert eliminate irrelevant predictors
  - Summary statistics - Frequency and correlation plots

Data Mining for Business Analytics: Concepts, Techniques, and Applications in R by GalitShmueli, Peter C. Bruce, InbalYahav, Nitin R. Patel, Kenneth C. LichtendahlJr. (2018)

FEATURE SELECTION - MULTICOLLINEARITY

- Multicollinearity occurs when **one predictor variable** in a model can be linearly predicted with other **predictor variables**.



$$y = \beta_0 + \beta_1 \boxed{x_1} + \beta_2 \boxed{x_2} + \ \dots \ + \beta_p \boxed{x_p} + \varepsilon$$

FEATURE SELECTION - MULTICOLLINEARITY

- Consequences of this issue includes:
  - Loss of precision
  - R^2 value takes on a high value despite not being statistically significant
  - Some of the signs of the coefficient might change
- Remedies to this problem includes:
  - Dropping problem variables *(selecting which feature to drop)*
  - Remedy using domain expertise
  - Get more data

FEATURE SELECTION - MULTICOLLINEARITY INTUITION

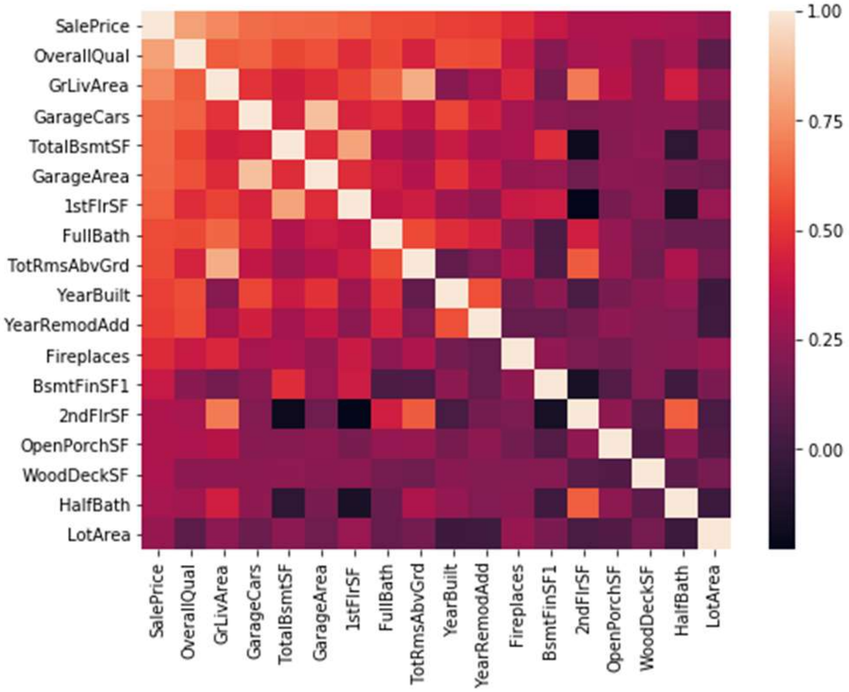
- Given the following dataset, there is a clear multicollinearity between the two predictor variables

House (#)	Price ($)	Price per square **foot** ($)	Price per square **metre** ($)
1	10000	80	240
2	25000	150	450
3	12000	75	225
4	8000	60	180
5	1500	100	300
6	2000	150	450

$$Price = \beta_0 + \beta_1 \, SQFT + \beta_2 \, SQM + \varepsilon$$

FEATURE SELECTION - DETECTING MULTICOLLINEARITY

- You can detect the correlation between variables using a correlation matrix (Rule of thumb: correlation coefficients > 0.8 signals multicollinearity)



FEATURE SELECTION - DETECTING MULTICOLLINEARITY

- Other ways of detecting multicollinearity include:
  - Variance Inflation Factor
  - Low Variance
  - `sklearn.feature_selection.f_regression`

SCIKIT LEARN EXAMPLE AND PRACTICE*

- Try out practice 2 in your in-class notebook
- Make sure you are able to identify the features which are correlated

THANK YOU!

HACKWAGON

• ACADEMY •