Machine Learning Prediction Models from Design to Deployment

Dr. Mohammed Al-Sarem
Department of Information Systems,
CCSE, Taibah University
2022


بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

A few words about “Al-Sarem M.”

Department of Information Systems,
College of Computer Science & Engineering,
Taibah University, Saudi Arabia

Address: Medina, KSA
msarem@taibahu.edu.sa/ mohsarem@gmail.com

Google Scholar:
https://scholar.google.com/citations?user=EG2h9d8AAAAJ&hl=en
ORCID: https://orcid.org/0000-0001-7172-8224
Publons: https://publons.com/researcher/AAL-1785-2020/
Linkedin: https://www.linkedin.com/in/mohammed-al-sarem-a9528b46/
Web of Science Researcher ID: AAL-1785-2020
Research Gate: https://www.researchgate.net/profile/Mohammed-Al-Sarem

A few words about “Al-Sarem M.” Cont’d

MOHAMMED AL-SAREM received the M.S. degree in information technology from the Faculty of Informatics and Computer Engineering, Volgograd State Technical University, Volgograd, Russia, and the Ph.D. degree from the Faculty of Informatics, University of Hassan II Casablanca-Mohamadia, Mohamadia, Morocco, in 2007 and 2014, respectively.
Associate Professor in the Information Systems Department, Taibah University, Medina, Saudi Arabia.
Published several articles and participated in managing several international conferences.
Current research interests include group decision making, multi-criteria decision making, data mining, deep learning, Adaptive e-learning, Natural Language Processing, and social analysis
https://scholar.google.com/citations?hl=en&user=EG2h9d8AAAAJ&view_op=list_works&sortby=pubdate

Machine Learning Prediction Models from Design to Deployment

Prerequisites

- No prior knowledge is required.
- At least, undergrad-level courses
  - Although no experience with pandas is required, basic knowledge of pandas will be beneficial.
  - All code is explained so that if you are new to pandas may follow along.
  - Installing Anaconda/ PyCharm
  - You can use any Online Editor: Jupyter, Colab

Interactions

- I will ask quick questions!
  - ❖ Use reactions to answer
- Several polls every lecture
- Please give reactions
  - ❖ at any time ☺
- Ask questions rising your hand

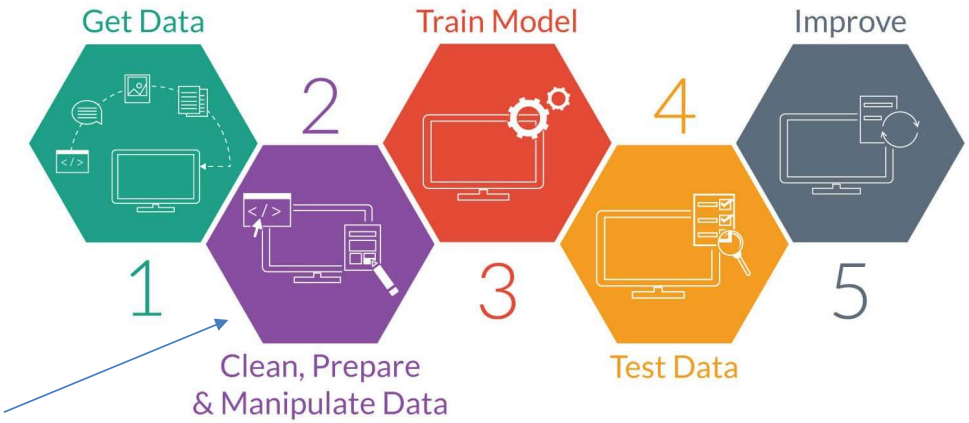
PLEASE …

Day 1: Content

- What is Machine learning?
- Installing environment
- Wrangling data
- Predicting regression: Introduction to Linear Regression

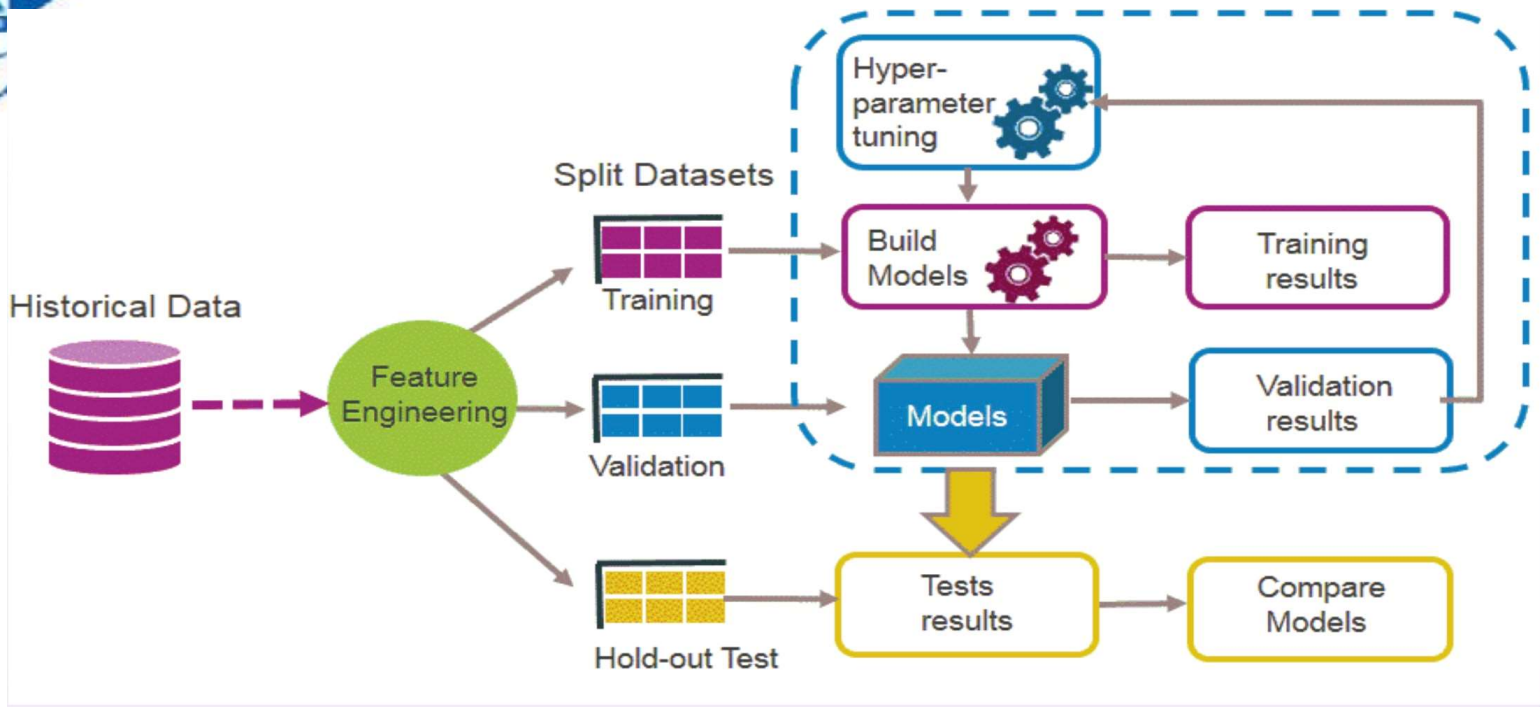
What is machine learning?

- Machine learning (ML) is:
  - ❖ The ability of computers to learn from data.
  - ❖ In practice, it means implementing computer algorithms whose weights are adjusted when new data comes in



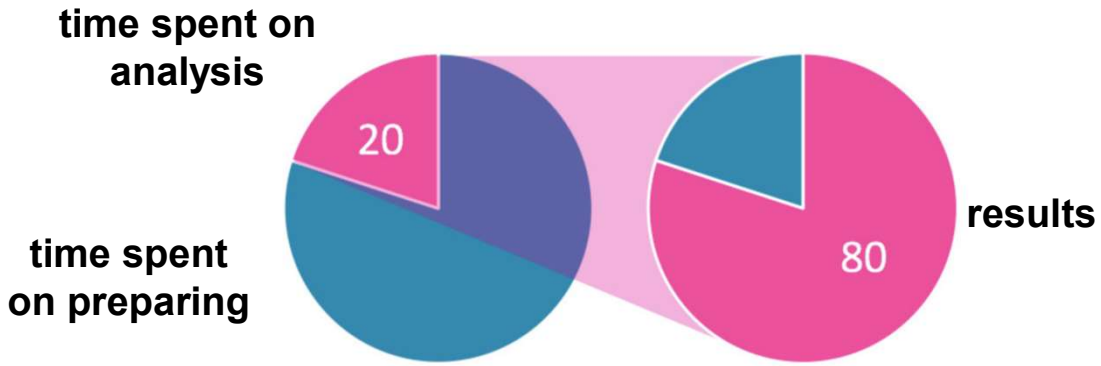
Data wrangling

Machine Learning Flow

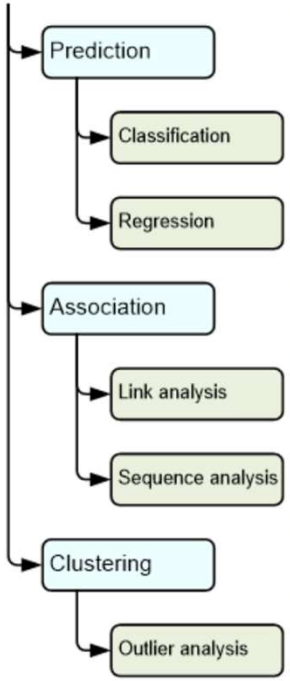


80/20 rule

- **80 percent of a data scientist's time is spent on finding, cleaning, preprocessing, and organizing data, leaving only 20 percent to actually perform analysis.**
- **However, the 20 percent effort determines 80% of the results.**



A Taxonomy of Tasks



Learning Method	Popular Algorithms
Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
Supervised	Decision trees, ANN/MLP, SVM, Rough sets, Genetic Algorithms
Supervised	Linear/Nonlinear Regression, Regression trees, ANN/MLP, SVM
Unsupervised	Apriory, OneR, ZeroR, Eclat
Unsupervised	Expectation Maximization, Apriory Algorithm, Graph-based Matching
Unsupervised	Apriory Algorithm, FP-Growth technique
Unsupervised	K-means, ANN/SOM
Unsupervised	K-means, Expectation Maximization (EM)

Data wrangling

Dr. Mohammed Al-Sarem 2022

Data wrangling?

- Data wrangling is:
  - ❖ a comprehensive term that encompasses the various stages of data preprocessing before machine learning can begin.
- It includes:
  - ❖ Data loading
  - ❖ Data cleaning
  - ❖ Data analysis
  - ❖ Data manipulation
  - ❖ ….

Connect to Google Colab and Google Driver

- Local file system
  - Uploading files from your local file system
    - **files.upload** returns a dictionary of the files which were uploaded.
    - The dictionary is keyed by the file name and values are the data which were uploaded.

```
from google.colab import files
uploaded = files.upload()
for fn in uploaded.keys():
  print('User uploaded file "{name}" with l
ength {length} bytes'.format(name=fn, lengt
h=len
                              (uploaded[fn])))
```

Connect to Google Colab and Google Driver

- Get file from Google Drive
  - You can access files in Drive in a number of ways, including:
    - Mounting your Google Drive in the runtime's virtual machine
    - Using a wrapper around the API such as PyDrive
    - Using the native REST API
- Mounting Google Drive locally
  - The easiest way
  - Mount Drive programmatical


```
from google.colab import drive
drive.mount('/content/drive')
```

```
%cd drive/MyDrive
%ls /content/drive/MyDrive/'Colab Notebooks
'/
```

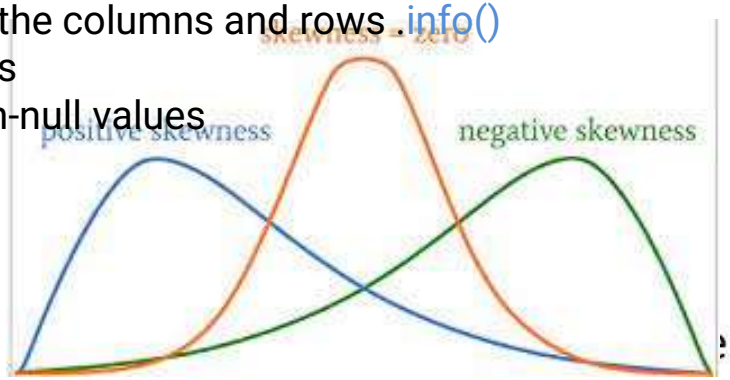
Data wrangling: Dataset 1 – Bike rentals

- Get dataset from: the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php)
- bike rentals dataset has been adjusted from the original dataset (https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset)
  - ❖ There are null values so that you can gain practice in correcting them

Data Set Characteristics:	Univariate	**Number of Instances:**	17389	**Area:**	Social
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	16	**Date Donated**	2013-12-20
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	704896

Working with DataFrame: First impression

- Understanding the data
  - Display the data using .head()
    - to view the data to ensure that it has loaded correctly
    - to display the first 5 rows of the DataFrame
    - You may place any positive integer in parentheses to view any number of rows.
    - What about the last rows in dataframe??
  - Numerical statistics may be viewed by using .describe()
    - You may need to scroll to the right to see all of the columns.
    - Comparing the mean and median (50%) gives an indication of **skewness**
  - Displays general information about the columns and rows .info()
    - To show #.of rows and columns
    - To show column types, and non-null values
    - Memory usage??



Correcting null values

- Process of correcting the null values
  - Finding the number of null values

```
# Sum null values
df_bikes.isna().sum().sum()
```

  - Displaying null values

```
# Show null value in df_bikes
df_bikes[df_bikes.isna().any(axis=1)]
```

  - Correcting null values:
    - Replacing with the median/mean
    - Groupby with the median/mean
      - Choices for the aggregate include .sum(), .count(), .mean(), and .median().

Tip:
(i) best practices, copy the original datafram;
(ii) The median is often a better choice than the mean. Why??

Obtaining the median/mean from specific rows

- In some cases, it may be advantageous to replace null values with data from specific rows.
  - When correcting temperature, aside from consulting historical records, taking the mean temperature of the day before and the day after should give a **good estimate**.
    - ❖ To find null values of the any column, enter the following code:

```
dataFrameName[dataFrameName['ColumnName'].isna()

    ''' Find the mean of raws of the day
    before, and the day after'''

    mean_temp = (df_bikes.iloc[700]['temp']
    df_bikes.iloc[702]['temp'])/2
```

    - ❖ Replace the null values:

```
df_bikes['temp'].fillna((mean_temp),
inplace=True)
```

Extrapolate dates

- Check the data type of the date columns
  - Refer to **.info( )** method to get the data type of all the attributes
  - The Object type commonly represented as a **string**
  - Date objects such as years and months must be extrapolated from **datetime types**
  - Use **pd.to_datetime ()** to convert to datetime type.

```
df_bikes['dteday'] =
pd.to_datetime(df_bikes['dteday'],infer_datetime_format=Tru
e)
```

- `infer_datetime_format=True` allows pandas to decide the kind of datetime object to store, a safe option in most cases.

Extrapolate dates

- Check the data type of the date columns
  - Convert month using the right method as follows:

```
# Import datetime
import datetime as dt
df_bikes['mnth'] = df_bikes['dteday'].dt.month
# Show last 5 rows
df_bikes.tail()
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
726	727	2012-12-27	1.0	1.0	12	0.0	4.0	1.0	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
727	728	2012-12-28	1.0	1.0	12	0.0	5.0	1.0	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
728	729	2012-12-29	1.0	1.0	12	0.0	6.0	0.0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
729	730	2012-12-30	1.0	1.0	12	0.0	0.0	0.0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
730	731	2012-12-31	1.0	NaN	12	0.0	1.0	0.0	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

What is this??

Extrapolate dates

- Check the data type of the date columns
  - You can use the .loc method to fill in the correct value. The .loc ( ) method is used to locate entries by row and column as follows:

```
# Change row 730, column 'yr' to 1.0
df_bikes.loc[730, 'yr'] = 1.0
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
726	727	2012-12-27	1.0	1.0	12	0.0	4.0	1.0	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
727	728	2012-12-28	1.0	1.0	12	0.0	5.0	1.0	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
728	729	2012-12-29	1.0	1.0	12	0.0	6.0	0.0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
729	730	2012-12-30	1.0	1.0	12	0.0	0.0	0.0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
730	731	2012-12-31	1.0	NaN	12	0.0	1.0	0.0	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

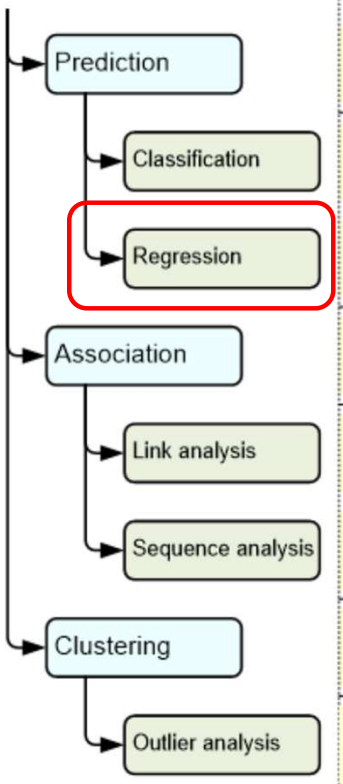
Data is normalized
What does this mean??

Deleting non-numerical columns

- For machine learning, all data columns:
  - Should be numerical.
  - Redundant should be removed. How can we detect that???
  - What if the data is textual???
  - Just Remove it.
    - Use .**drop( )** function.

```
df_bikes = df_bikes.drop('dteday', axis=1)
```

A Taxonomy of Tasks



Learning Method	Popular Algorithms
Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
Supervised	Decision trees, ANN/MLP, SVM, Rough sets, Genetic Algorithms
Supervised	Linear/Nonlinear Regression, Regression trees, ANN/MLP, SVM
Unsupervised	Apriory, OneR, ZeroR, Eclat
Unsupervised	Expectation Maximization, Apriory Algorithm, Graph-based Matching
Unsupervised	Apriory Algorithm, FP-Growth technique
Unsupervised	K-means, ANN/SOM
Unsupervised	K-means, Expectation Maximization (EM)

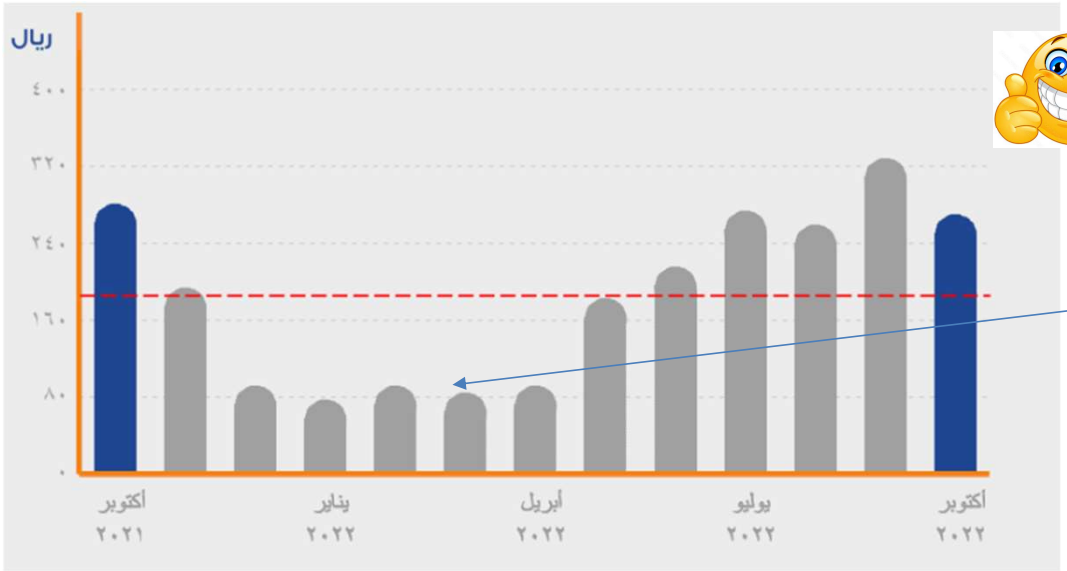
Linear Regression: A simple example: predicting electricity use

- What will bill amount of power consumption be in my house next month?
  - Difficult to build an “a priori” model from first principles to answer this question.
  - But, relatively easy to record past days of consumption, plus additional features that affect consumption (i.e., weather)

Date	High Temperature (C)	Peak Demand (KW per H)	Amount
04.10.2022	[37°-43°]	1360	٢٩٣,٠٢ ريال
05.09.2022	[43°-47°]	1682	٣٥٩,٦٧ ريال
04.07.2022	[44°-51°]	1293	٢٧٩,١٥ ريال
05.05.2022	[43°-49°]	1059	٢٣٠,٧١ ريال
…	…	…	…

Plot of month vs. temperature

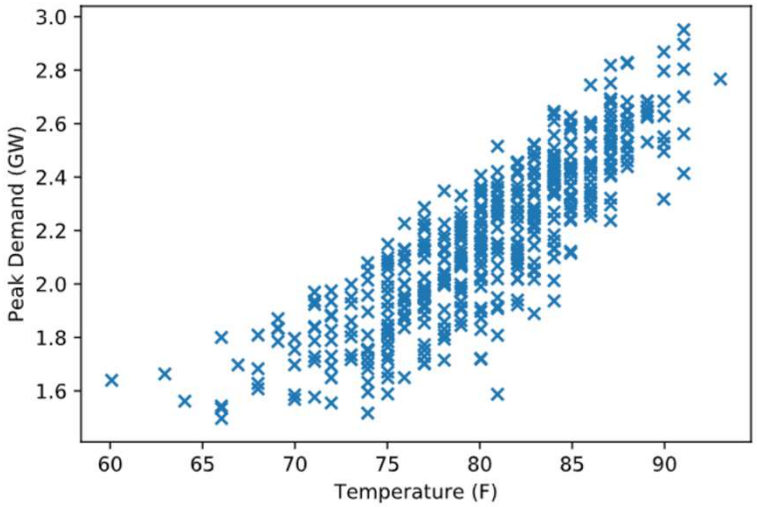
- Plot of Bill amount vs. Date for the last 1 year (October – October)

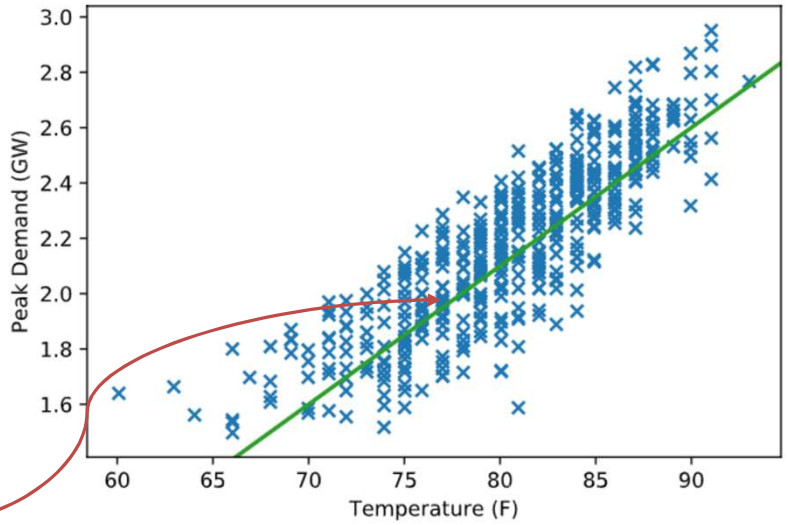


Clear: there is a correlation Btw the month of the year and the amount of consumption

Plot of consumption vs. temperature

- Plot of high temperature vs. peak demand for summer months (June – August) for past six years





Hypothesis: linear model

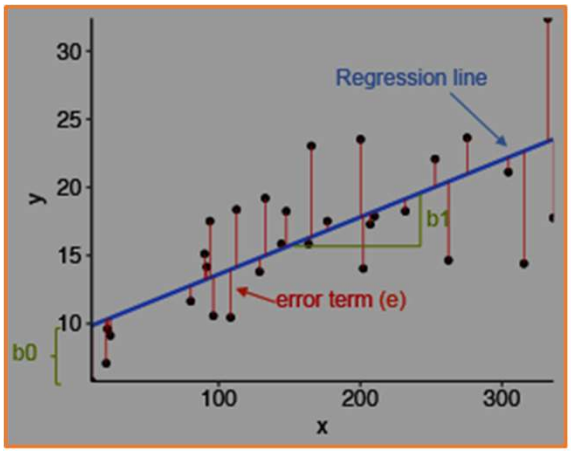
Let's suppose that the peak demand approximately fits a linear model.

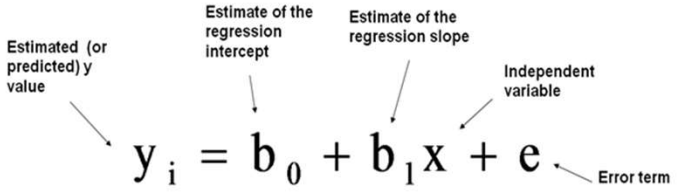
Hypothesis: linear model

Let's suppose that the peak demand approximately fits a linear model

$$\text{Peak\_Demand} \approx b_0 \cdot \text{High\_Temperature} + b_1$$

Here b_0 is the "slope" of the line, and b_1 = is the intercept



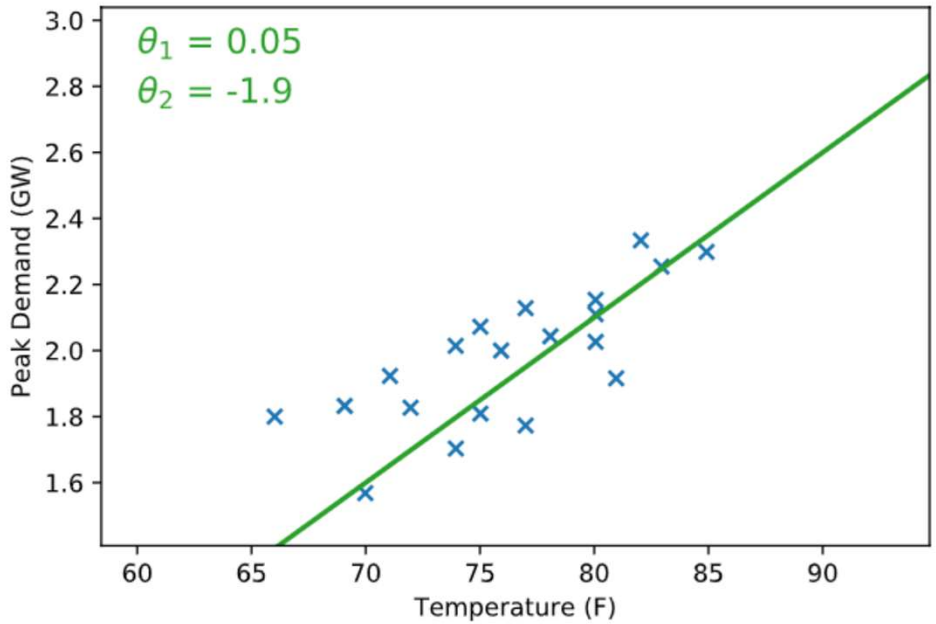


Making predictions

- Importantly, our model also lets us make predictions about new days
  - What will the peak demand be tomorrow?
- If we know the high temperature will be 72 degrees (ignoring for now that this is also a prediction), then we can predict peak demand to be:
  - Predicted_Peak_Demand = $b_0 \cdot 72 + b_1$ = 1.821 GW
- Equivalent to just "finding the point on the line"

Predicted output for each data point

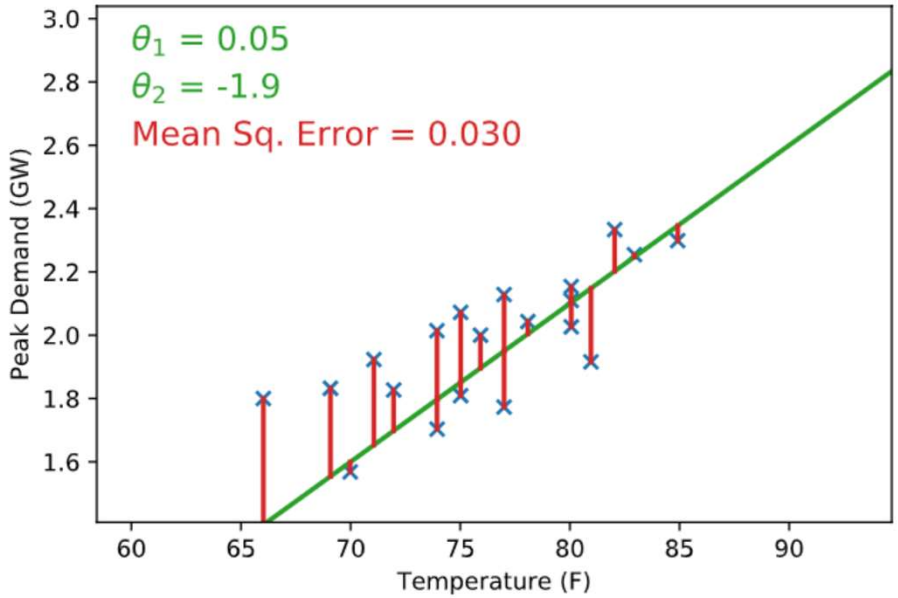
$$\text{Predicted\_Peak\_Demand}^{(i)} = \theta_1 \cdot \text{High\_Temperature}^{(i)} + \theta_2$$



Predicted output for each data point

$$\text{Predicted\_Peak\_Demand}^{(i)} = \theta_1 \cdot \text{High\_Temperature}^{(i)} + \theta_2$$

$$\text{Peak\_Demand}^{(i)}$$




1
POLL

1

Assume we have the following data: Find the regression equation that fit the data

Month	Spend	Sales
1	1000	9914
2	4000	40487
3	5000	54324
4	4500	50044
5	3000	34719
6	4000	42551
7	9000	94871
8	11000	118914
9	15000	158484
10	12000	131348
11	7000	78504
12	3000	36284

$$\beta_1 = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

1- Manual Calculation
2- Using Excel
3- Python code

Multiple Linear Regression

- Multiple or multivariate linear regression is a case of linear regression with two or more independent variables
  - For two independent variables, the estimated regression function is $f(x_1, x_2) = b_0 + b_1x_1 + b_2x_2$
  - The regression plane in a three-dimensional space.
- The case of more than two independent variables is similar, but more general.
  - The estimated regression function is $f(x_1, \ldots, x_r) = b_0 + b_1x_1 + \cdots + b_rx_r$, and there are $r + 1$ weights to be determined when the number of inputs is r.

Python Packages for Linear Regression

- NumPy Package:
  - a fundamental Python scientific package that allows many high-performance operations on single-dimensional and multidimensional arrays.
  - It also offers many mathematical routines
  - It is Open-source
- Scikit-learn Package:
  - widely used Python library for machine learning
  - built on top of NumPy and some other packages.
  - provides the means for preprocessing data, reducing dimensionality, implementing regression, classifying, clustering, and more.
  - It is Open-source
- Statsmodels Package
  - it's a powerful Python package for the estimation of statistical models, performing tests, and more.
  - It's open-source as well

Evaluation Metrics

- There are three error metrics that are commonly used for evaluating and reporting the performance of a regression model; they are:
  - Mean Squared Error (MSE).
    - the mean or average of the squared differences between predicted and expected target values in a dataset.
  - Root Mean Squared Error (RMSE).
    - RMSE, is an extension of the mean squared error.
    - Has the same units as the original units of the target value that is being predicted
  - Mean Absolute Error (MAE)
    - the average of the absolute error values
    - Does not consider direction

Evaluation Metrics

- Mean Squared Error (MSE).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

- Root Mean Squared Error (RMSE).

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

- Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - x|$$

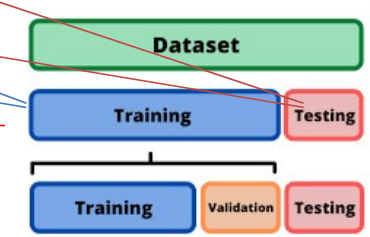
Regression Analysis in Scikit-learn library

Scikit-learn

```
# Import train_test_split
from sklearn.model_selection import train_test_split

# Import Linear Regression
from sklearn.linear_model import LinearRegression

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y
, random_state=2)
```



Regression Analysis in Scikit-learn library

Scikit-learn

```
# Initialize LinearRegression model
lin_reg = LinearRegression()
# Fit lin_reg on training data
lin_reg.fit(X_train, y_train)
# Predict X_test using lin_reg
y_pred = lin_reg.predict(X_test)
# Import mean_squared_error
from sklearn.metrics import mean_squared_error
# Import numpy
import numpy as np
# Compute mean_squared_error as mse
mse = mean_squared_error(y_test, y_pred)
# Compute root mean squared error as rmse
rmse = np.sqrt(mse)
# Display root mean squared error
print("RMSE: %0.2f" % (rmse))
```

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

Linear Regression: Model Evaluation

Dr. Mohammed Al-Sarem 2022

Dr. Mohammed Al-Sarem 2022

Thank You!

Dr. Mohammed Al-Sarem 2022