DataWrangler

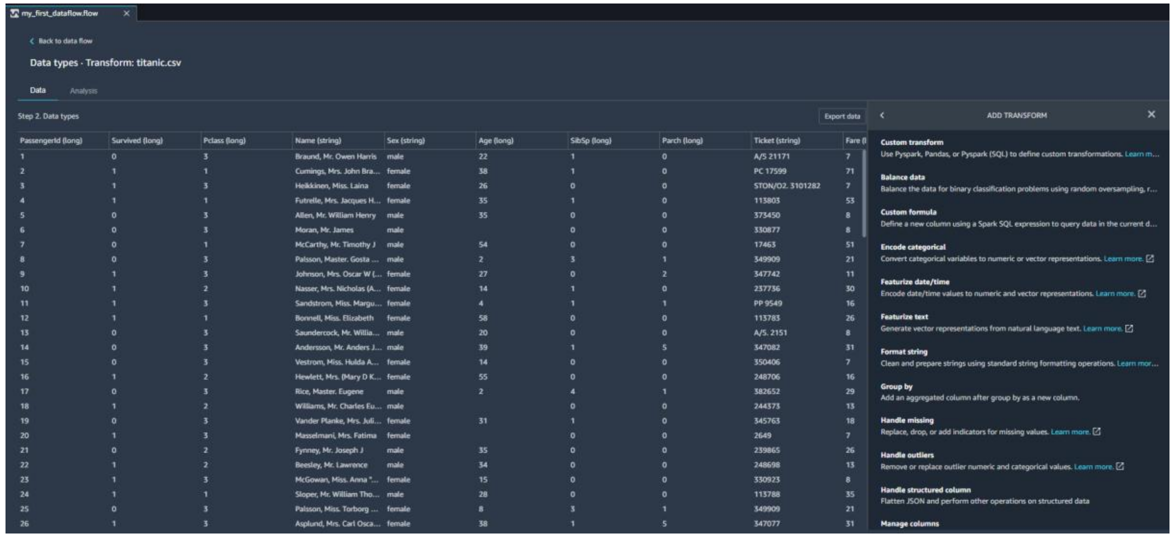
Project Overview

- In this project, we will leverage the power of data Wrangler service in AWS to prepare, clean and visualize the data.
- We will analyze the Titanic dataset which contains features related to Titanic passengers and cardiovascular disease datasets (final project).
- Here are the key learning outcomes:
  - Understand feature engineering strategies and tools.
  - Understand the fundamentals of Data Wrangler in AWS.
  - Perform one hot encoding and normalization.
  - Perform data visualization Using Data Wrangler.
  - Export a data wrangler workflow into Python Script.
  - Create a custom formula and apply it to a given column in the data.
  - Generate summary table tables in Data Wrangler.
  - Generate bias reports.

Sagemaker Data Wrangler 101

Data Wrangler 101

- Amazon SageMaker Data Wrangler accelerates the process of data preparation, exploration, cleaning, visualization and feature engineering. It makes creating Extract, Transform and Load (ETL) pipelines much easier.



Data types · Transform: titanic.csv

Step 2. Data types

PassengerId (long)	Survived (long)	Pclass (long)	Name (string)	Sex (string)	Age (long)	SibSp (long)	Parch (long)	Ticket (string)	Fare (l
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7
2	1	1	Cumings, Mrs. John Bra...	female	38	1	0	PC 17599	71
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7
4	1	1	Futrelle, Mrs. Jacques H...	female	35	1	0	113803	53
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8
6	0	3	Moran, Mr. James	male		0	0	330877	8
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51
8	0	3	Palsson, Master. Gosta ...	male	2	3	1	349909	21
9	1	3	Johnson, Mrs. Oscar W (...	female	27	0	2	347742	11
10	1	2	Nasser, Mrs. Nicholas (A...	female	14	1	0	237736	30
11	1	3	Sandstrom, Miss. Margu...	female	4	1	1	PP 9549	16
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26
13	0	3	Saundercock, Mr. Willia...	male	20	0	0	A/5. 2151	8
14	0	3	Andersson, Mr. Anders J...	male	39	1	5	347082	31
15	0	3	Vestrom, Miss. Hulda A...	female	14	0	0	350406	7
16	1	2	Hewlett, Mrs. (Mary D K...	female	55	0	0	248706	16
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29
18	1	2	Williams, Mr. Charles Eu...	male		0	0	244373	13
19	0	3	Vander Planke, Mrs. Juli...	female	31	1	0	345763	18
20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7
21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26
22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13
23	1	3	McGowan, Miss. Anna "...	female	15	0	0	330923	8
24	1	1	Sloper, Mr. William Tho...	male	28	0	0	113788	35
25	0	3	Palsson, Miss. Torborg ...	female	8	3	1	349909	21
26	1	3	Asplund, Mrs. Carl Osca...	female	38	1	5	347077	31

ADD TRANSFORM

Custom transform
Use Pyspark, Pandas, or Pyspark (SQL) to define custom transformations. Learn m...

Balance data
Balance the data for binary classification problems using random oversampling, r...

Custom formula
Define a new column using a Spark SQL expression to query data in the current d...

Encode categorical
Convert categorical variables to numeric or vector representations. Learn more.

Featurize date/time
Encode date/time values to numeric and vector representations. Learn more.

Featurize text
Generate vector representations from natural language text. Learn more.

Format string
Clean and prepare strings using standard string formatting operations. Learn mor...

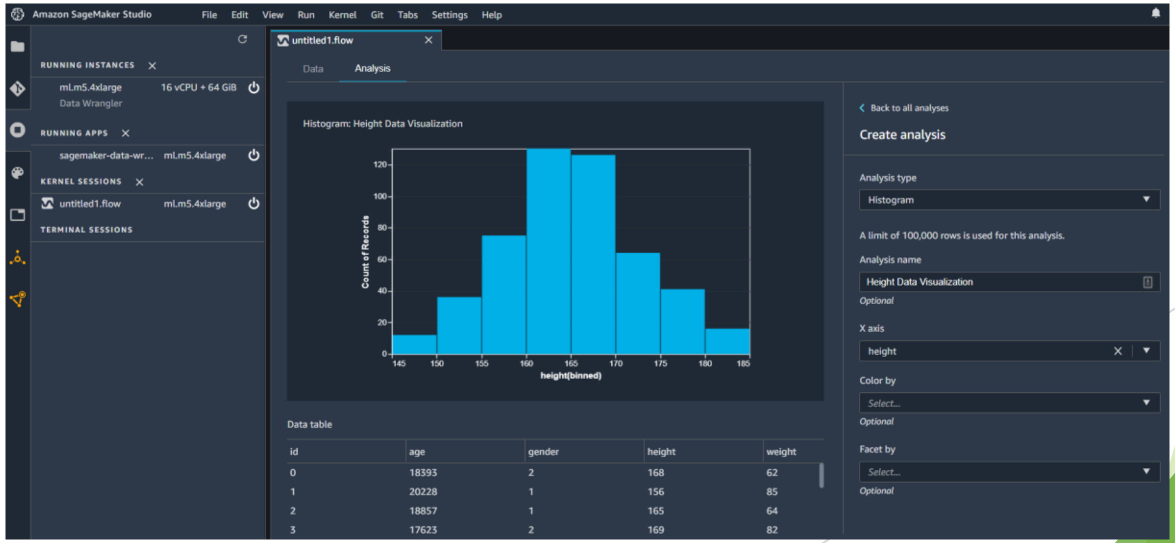
Group by
Add an aggregated column after group by as a new column.

Handle missing
Replace, drop, or add indicators for missing values. Learn more.

Handle outliers
Remove or replace outlier numeric and categorical values. Learn more.

Handle structured column
Flatten JSON and perform other operations on structured data

Manage columns



RUNNING INSTANCES

ml.m5.4xlarge Data Wrangler	16 vCPU + 64 GiB	

RUNNING APPS

sagemaker-data-wr...	ml.m5.4xlarge

KERNEL SESSIONS

untitled1.flow	ml.m5.4xlarge

TERMINAL SESSIONS

Data table

id	age	gender	height	weight
0	18393	2	168	62
1	20228	1	156	85
2	18857	1	165	64
3	17623	2	169	82

Create analysis

Analysis type: Histogram

A limit of 100,000 rows is used for this analysis.

Analysis name: Height Data Visualization
Optional

X axis: height

Color by: *Select...*
Optional

Facet by: *Select...*
Optional

Data Wrangler 101

- SageMaker Data Wrangler is **cloud based** and **doesn’t require any code**!
- Data can be imported into data Wrangler from more than one source such as **S3**, RedShift and SageMaker Feature Store.
- Data could be in **CSV, database tables and Parquet** formats.
- Data Wrangler includes over **300 data transformations** such as one hot encoding, normalization, imputation of missing data, ..etc.
- Several data visualization templates are available to generate **bar charts, line plots, histograms, scatterplots**…etc.
- Data Transformation **workflows can be exported** from data wrangler to a notebook or script so it can be automated with SageMaker Pipelines.
- Check out success stories from customers: https://aws.amazon.com/sagemaker/data-wrangler/

What Is Feature Engineering?

- Machine Learning algorithms require training data to train.
- Feature engineering is a critical task that is performed by data scientists prior to training AI/ML models to ensure solid trained model performance.
- Feature engineering is an art of introducing new features that weren’t existing before.
- Data scientists spend 80% of their time performing feature engineering.
- The remaining 20% is the easy part which includes training the model and performing hyperparameters optimization.
- As a data scientist, you may need to:
    1. Highlight important information in the data
    2. Remove/isolate unnecessary information (e.x.: outliers).
    3. Add your own expertise and domain knowledge to the alter the data.

Photo Credit: https://pixabay.com/illustrations/network-data-memory-data-collection-4478146/

Feature Engineering: Proper Questions To Ask?

- As a data scientist, you need to answer the following questions:

Which features should I select?
Can I add my domain knowledge to use less features?
Can I come up with new features from the data I have at hand?
What should I put in the missing data locations?
What are the capabilities of the ML model I have?

It is important to choose features that are most relevant to the problem.

Adding new features that are unnecessary will increase the computational requirements needed to train the model (curse of dimensionality).

There are many techniques that could be used to reduce the number of features (compress/encode the data) such as Principal Component Analysis (PCA) – will be covered later.

FEATURE ENGINEERING: QUIZ

- Let’s take a look at this data and see what’s wrong with it!

CUSTOMER ID	CUSTOMER NAME	LOCATION	CLICK ON AD?
1	Georgina	USA	Yes
2	Leila	Canada	1
3	Sarah	France	0
4	Bird		1
5	Max	Netherlands	0
6	Sarah	France	0

Feature Engineering: Solution

- Let’s take a look at this data and see what’s wrong with it!

CUSTOMER ID	CUSTOMER NAME	LOCATION	CLICK ON AD?
1	Georgina	USA	Yes
2	Leila	Canada	1
3	Sarah	France	0
4	Bird		1
5	Max	Netherlands	0
6	Sarah	France	0

ENTIRE COLUMN REQUIRES ENCODING

MISSING INFORMATION

REQUIRES FORMATTING

DUPLICATE ENTRY

FEATURE ENGINEERING TECHNIQUES

- Imputation
- Handling Outliers
- Binning
- Log Transform
- One-Hot Encoding
- Feature Split
- Scaling

Feature Engineering: Tools

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.

JUPYTER NOTEBOOKS

AMAZON SAGEMAKER DATA WRANGLER

AWS GLUE

ONE-HOT ENCODING

One-hot Encoding: Why Do We Need It?

- Can we simply replace colors with integer values?
- The machine learning model will assume that:

$$\boldsymbol{GREEN > YELLOW > RED}$$

COLOR
RED
RED
YELLOW
GREEN
YELLOW

WRONG!

ENCODED COLOR
1
1
2
3
2

One-hot Encoding

- One hot encoding is widely used in machine learning.
- It works by converting values such as “color” into columns with 1’s and 0’s in them.
- Since machine learning models deal with numbers, we perform one hot encoding to convert from categorical data into numerical.
- If you have N categories, you will need N-1 binary columns to represent them.

COLOR
RED
RED
YELLOW
GREEN
YELLOW

RED	YELLOW	GREEN
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

ONE-HOT ENCODING: ORDINAL Vs. Nominal

- The difference between nominal and ordinal data is as follows:
  - In ordinal data, order is important.
  - In nominal data, order is not important.

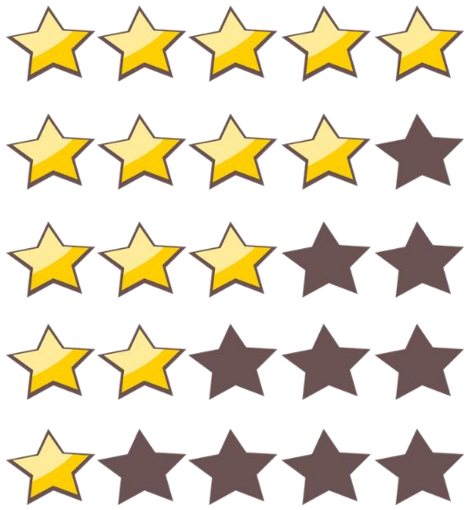
NOMINAL

Order of colors doesn't mean anything!

COLOR
RED
RED
YELLOW
GREEN
YELLOW

ORDINAL

Order is important!



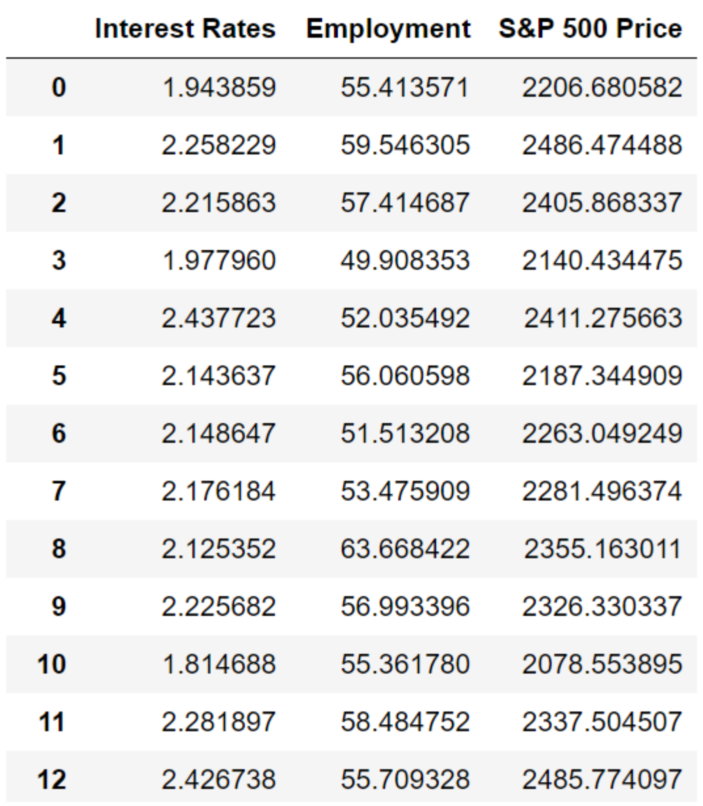
- *1 star means poor quality course*
- *5 star means great quality course*

Photo Credit: https://pixabay.com/vectors/rating-stars-system-evaluation-153125/

Feature Scaling

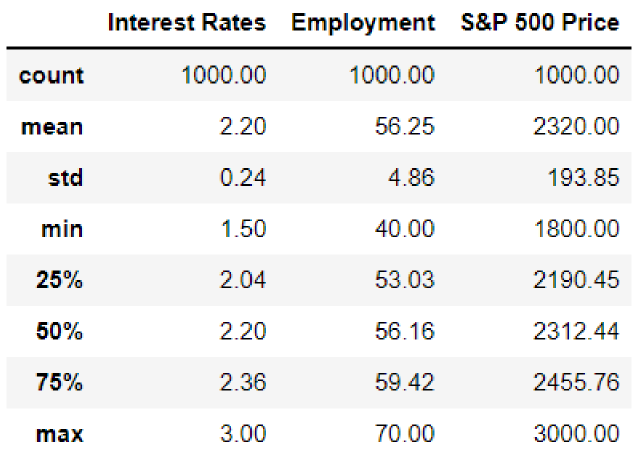
- Feature Scaling is an important step to take prior to training of machine learning models to ensure that features are within the same scale.
- Example: interest rate and employment score are at a different scale. This will result in one feature dominating the other feature.
- Scikit Learn offers several tools to perform feature scaling.

RAW ORIGINAL DATASET



	Interest Rates	Employment	S&P 500 Price
0	1.943859	55.413571	2206.680582
1	2.258229	59.546305	2486.474488
2	2.215863	57.414687	2405.868337
3	1.977960	49.908353	2140.434475
4	2.437723	52.035492	2411.275663
5	2.143637	56.060598	2187.344909
6	2.148647	51.513208	2263.049249
7	2.176184	53.475909	2281.496374
8	2.125352	63.668422	2355.163011
9	2.225682	56.993396	2326.330337
10	1.814688	55.361780	2078.553895
11	2.281897	58.484752	2337.504507
12	2.426738	55.709328	2485.774097

QUICK STATS!

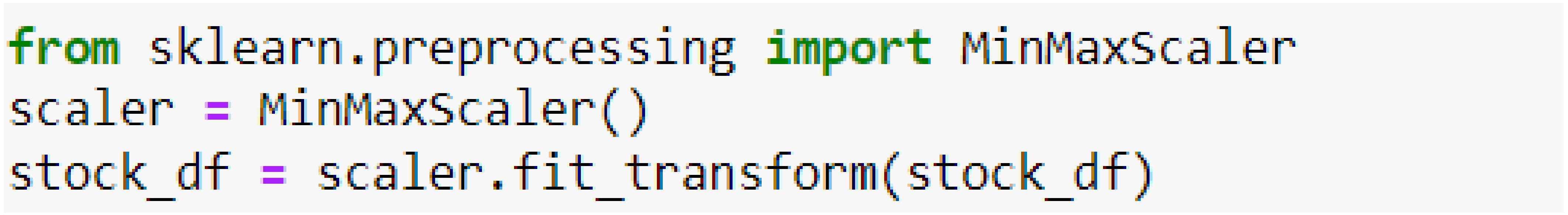


	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00

Normalization

- Normalization is conducted to make feature values range from 0 to 1.

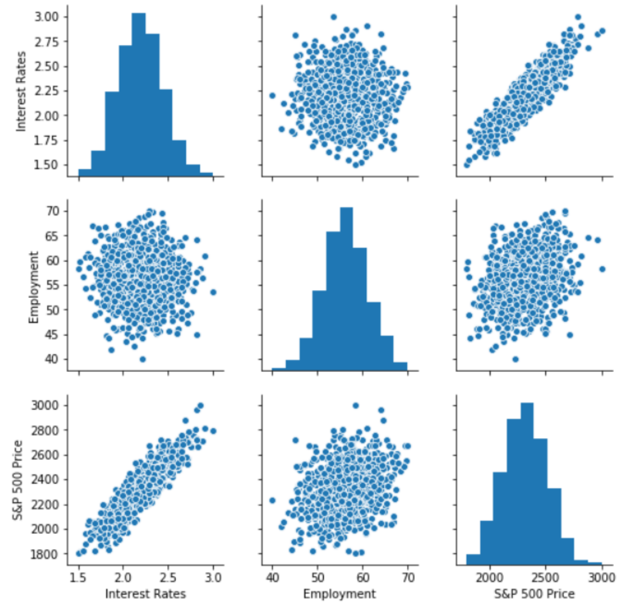
$$\boxed{x' = \frac{x - \min(x)}{\max(x) - min(x)}}$$

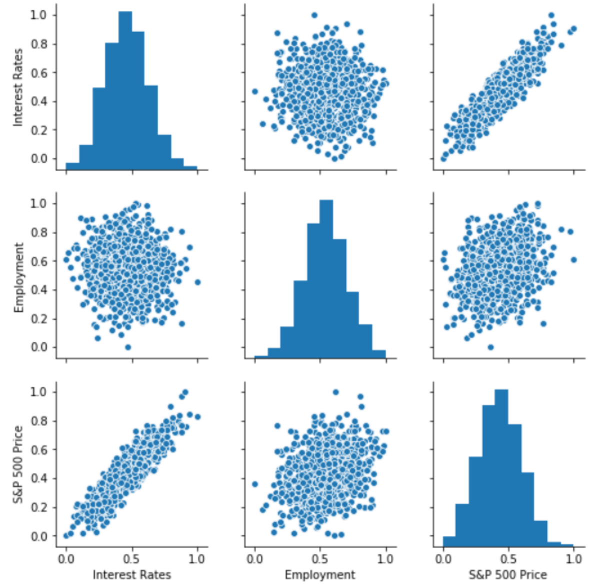


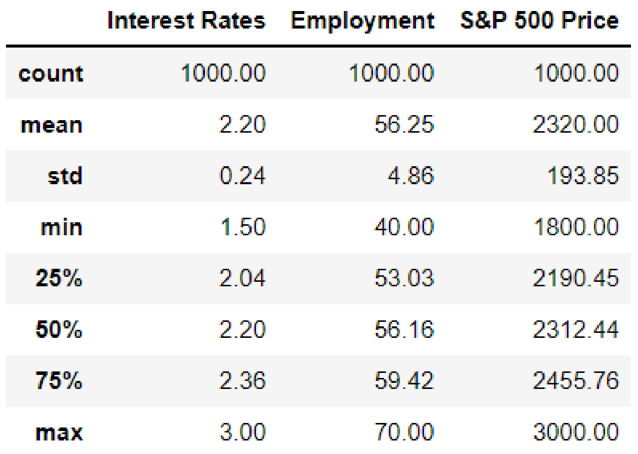
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
stock_df = scaler.fit_transform(stock_df)
```

Normalization

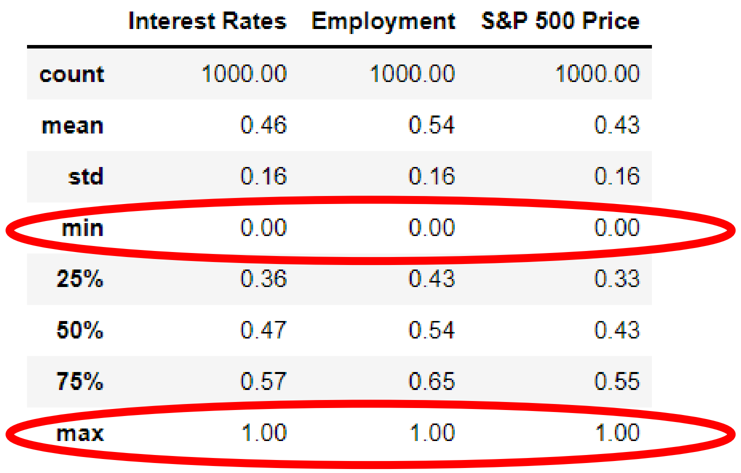
- Normalization is conducted to make feature values range from 0 to 1.







	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00



	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	0.46	0.54	0.43
std	0.16	0.16	0.16
min	0.00	0.00	0.00
25%	0.36	0.43	0.33
50%	0.47	0.54	0.43
75%	0.57	0.65	0.55
max	1.00	1.00	1.00

Normalization

$$x' = \frac{x - \min(x)}{\max(x) - min(x)} = \frac{2206 - 1800}{3000 - 1800} = 0.338$$

	S&P 500 PRICE (ORIGINAL)	S&P 500 PRICE (NORMALIZED)
Randomly selected Value	2206	0.338
Average	2319	0.432
Maximum	3000	1
Minimum	1800	0

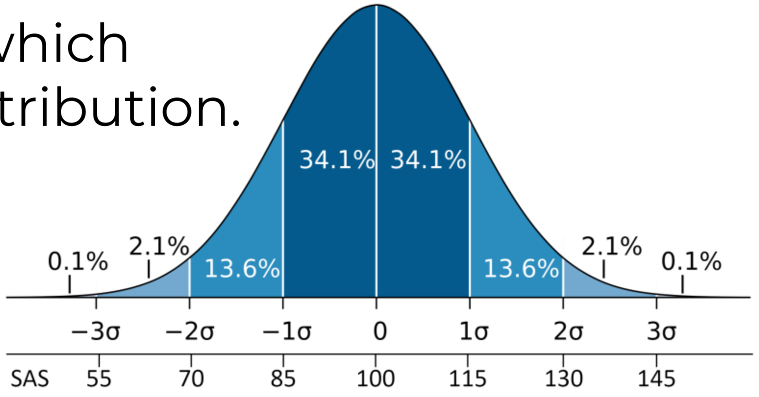
NORMALIZATION/ SCALING

- S&P 500 PRICE (Max) = 3000
- S&P 500 PRICE (Min) = 1800

NOTE THAT NUMBERS NOW RANGE FROM 0 TO 1 AFTER NORMALIZATION

Standardization

- Standardization is conducted to transform the data to have a mean of zero and standard deviation of 1.
- Standardization is also known as Z-score normalization in which properties will have the behaviour of a standard normal distribution.



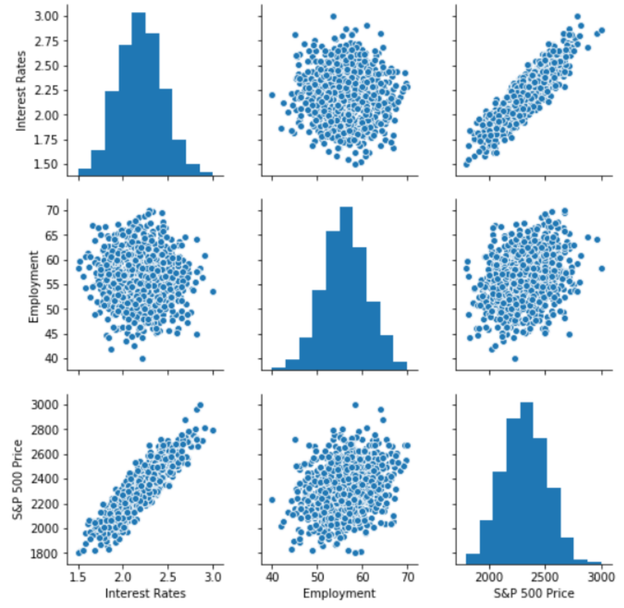
$$\boxed{z = \frac{x - \bar{x}}{\sigma}}$$

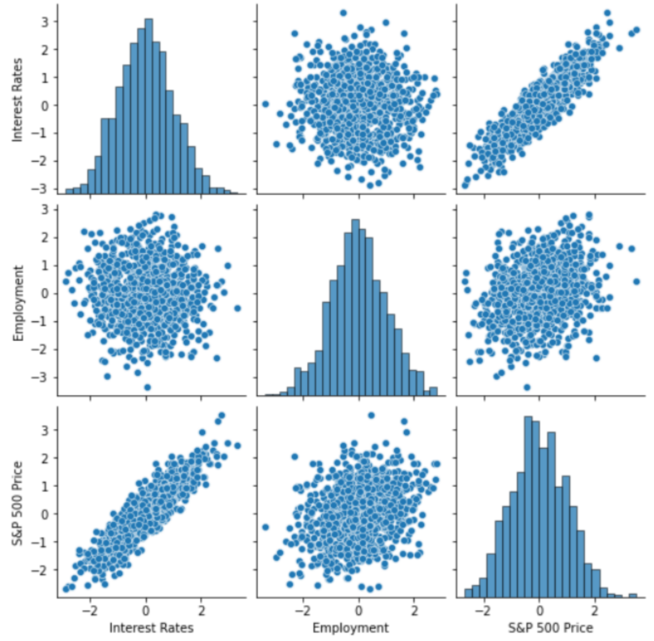
```python
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
stock_df = scaler.fit_transform(stock_df)
```

Image Source: https://commons.wikimedia.org/wiki/File:Wechsler.svg

Standardization

- Standardization transforms data to have a mean of zero and standard deviation of 1.





	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	2.20	56.25	2320.00
std	0.24	4.86	193.85
min	1.50	40.00	1800.00
25%	2.04	53.03	2190.45
50%	2.20	56.16	2312.44
75%	2.36	59.42	2455.76
max	3.00	70.00	3000.00

	Interest Rates	Employment	S&P 500 Price
count	1000.00	1000.00	1000.00
mean	0.00	0.00	-0.00
std	1.00	1.00	1.00
min	-2.88	-3.34	-2.68
25%	-0.66	-0.66	-0.67
50%	0.01	-0.02	-0.04
75%	0.68	0.65	0.70
max	3.33	2.83	3.51

Standardization

$$z = \frac{x - \overline{x}}{\sigma} = \frac{2206 - 2319}{193.8} = -0.583$$

	S&P 500 PRICE (ORIGINAL)
Randomly selected Value	2206
Average	2319
Maximum	3000
Minimum	1800

STANDARIZATION

$\xrightarrow{\hspace{3cm}}$

- S&P 500 PRICE (Mean) = 2319
- S&P 500 PRICE (Std) = 193.8

S&P 500 PRICE (STANDARIZED)
-0.583
0
3.513
-2.67

NOTE THAT AFTER STANDARDIZATION THE AVERAGE IS SET TO ZERO

Always Remember!

“A normalized dataset will always range from 0 to 1”

“A standardized dataset will always have a mean of 0 and standard deviation of 1, but can have any upper and lower values”

WHEN SHOULD I PERFORM STANDARDIZATION VS. NORMALIZATION?

Scaling (standardization or normalization) is required when we use any machine learning algorithm that require **gradient calculation.**

Examples of machine learning algorithms that require gradient calculations are: linear/logistic regression and artificial neural networks

Having different scales for each feature will result in a different step size which in turn jeopardizes the process of reaching a minimum point.

Scaling is not required for distance-based and tree-based algorithms such as K-Means Clustering, Support Vector Machines and K Nearest Neighbors, decision trees, random forest, and XG-Boost.

STANDARDIZATION Vs. Normalization?

Generally speaking, there is no right or wrong answer!

In case of neural networks, normalization is preferred since we don't assume any data distribution.

Standardization is preferred when data follows gaussian distribution

Standardization is preferred over normalization when there are a lot of outliers.

Thanks so much to

Professor Ryan Ahmed