SC310005 Artificial Intelligence

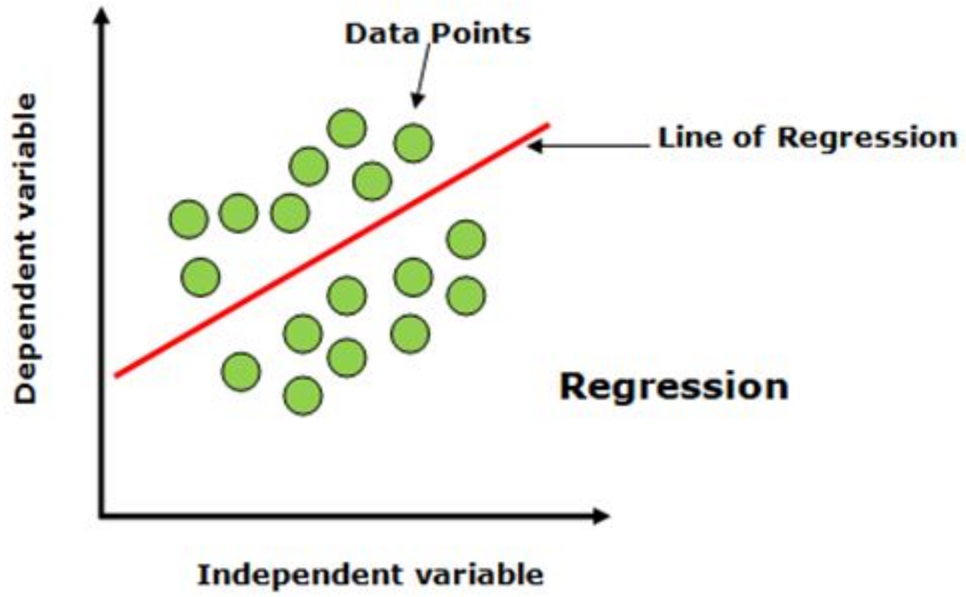
Lecture 7: Regression

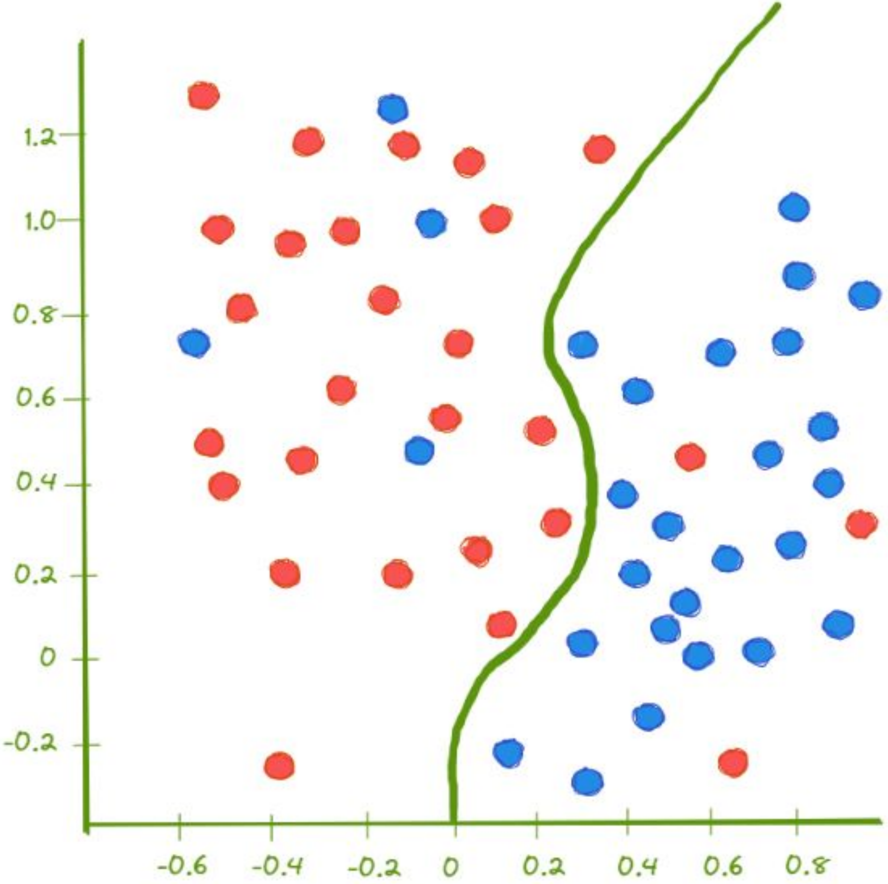
teerapong.pa@chula.ac.th

Reference

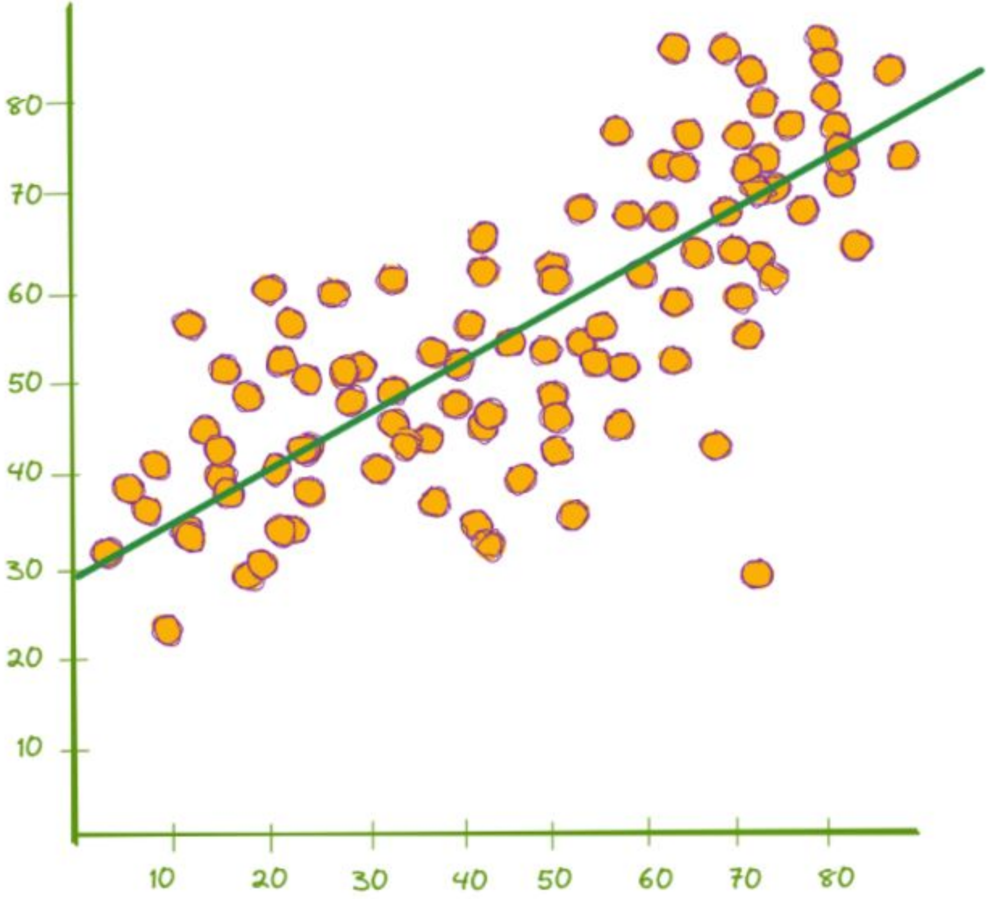
- https://www.linkedin.com/pulse/understanding-linear-regression-machine-learning-tool-jadhav
- https://medium.com/@rndayala/linear-regression-a00514bc45b0
- https://medium.com/@polanitzer/the-minimum-mean-absolute-error-mae-challenge-928dc081f031
- https://www.investopedia.com/terms/p/positive-correlation.asp
- https://www.linkedin.com/pulse/correlation-vs-causality-krishnakumar-ramanathan

Understanding Linear Regression in Machine Learning: A Fundamental Tool for Predictive Modeling



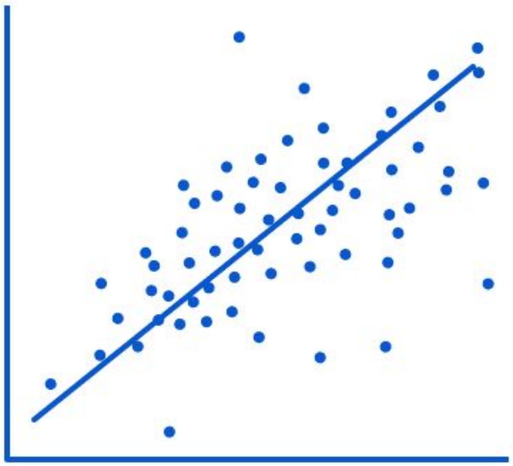


classification

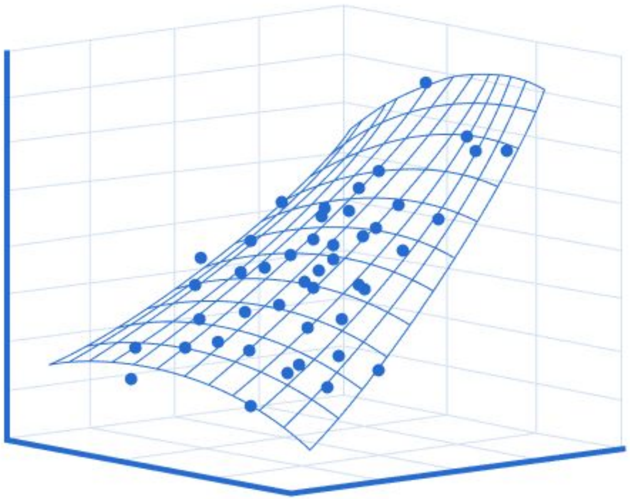


regression

Simple Linear Regression

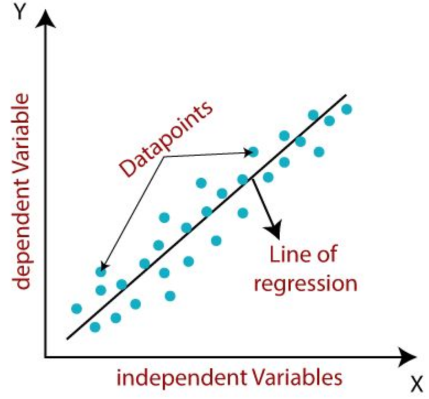


Multiple Linear Regression



mobidev

Linear Regression (Equation)



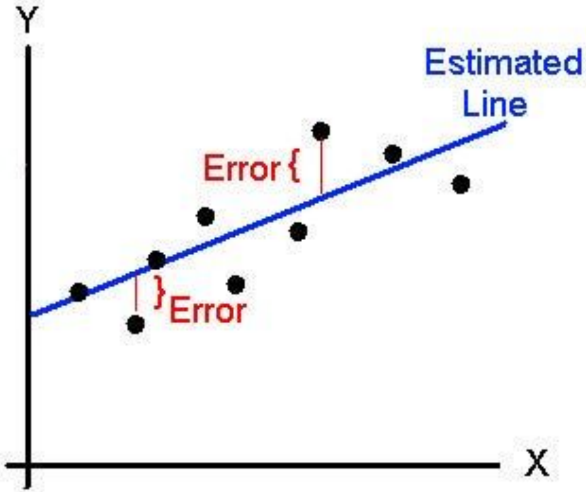
Estimated (or predicted) Y value for observation i

Estimate of the regression intercept

Estimate of the regression slope

Value of X for observation i

$$\hat{Y}_i = b_0 + b_1 X_i$$



single value of dependent variable

slope

single value of independent variable

y-intercept

$$y = mx + b$$

$$Y = \beta_0 + \beta_1 X + \epsilon$$

all observed values for dependent variable

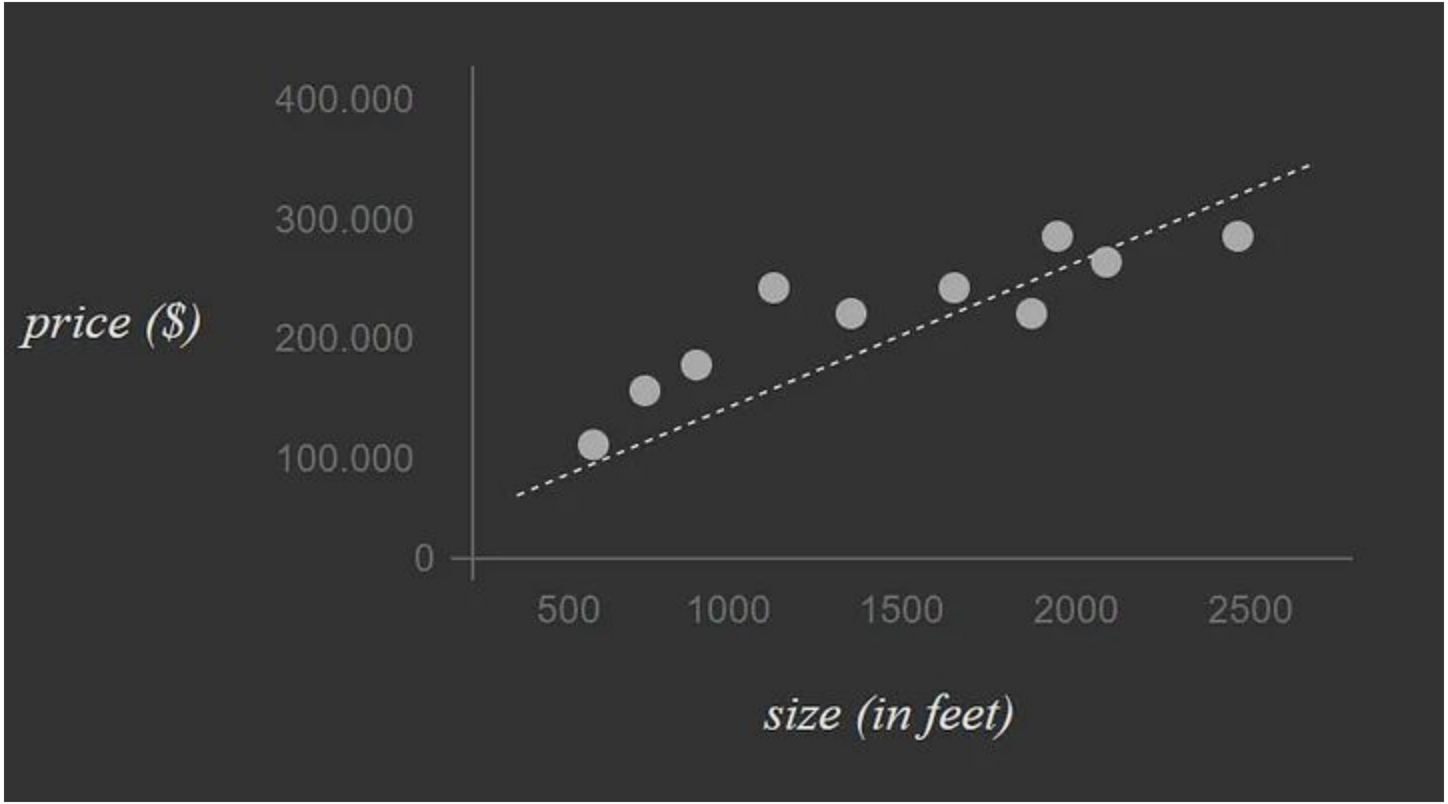
y-intercept a.k.a "bias"

slope a.k.a. "coefficient"

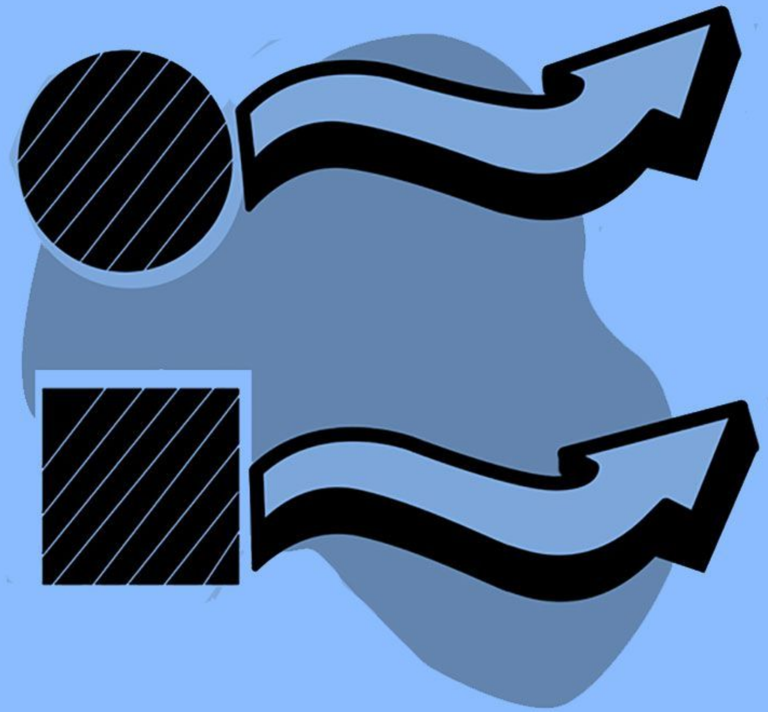
all observed values of independent variable

error*

* additional term


price ($)
400.000
300.000
200.000
100.000
0
500
1000
1500
2000
2500
size (in feet)

size (feet^2) (x)	price (\$) (y)
815	165,000
1510	310,000
2100	410,000
...	...

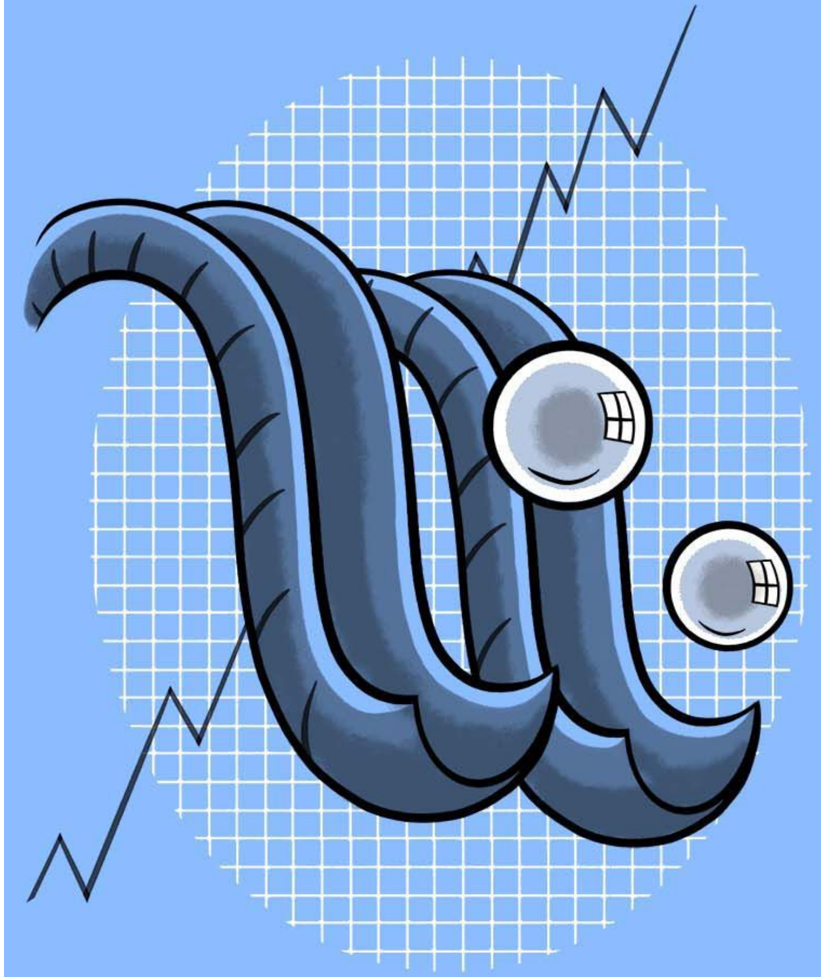


Correlation

[, kor-ə-'lā-shən]

A statistic that measures the degree to which two securities move in relation to each other.

Investopedia

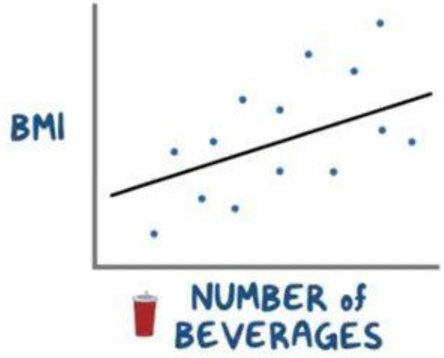


Positive Correlation

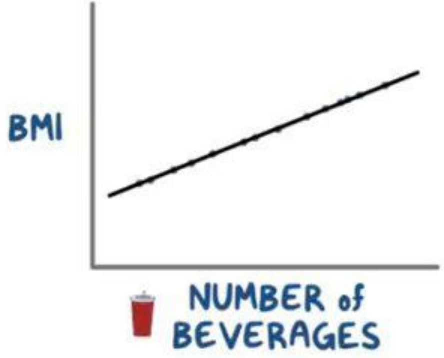
[ˈpä-zə-tiv ˌkȯr-ə-ˈlā-shən]

A relationship between two variables that move in the same direction.

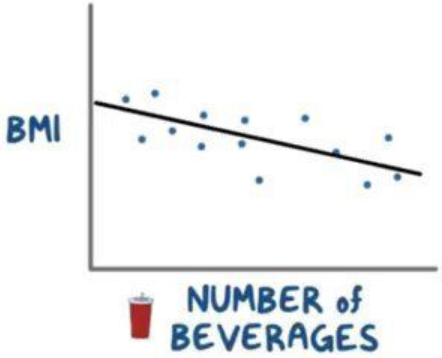
SCATTERPLOT



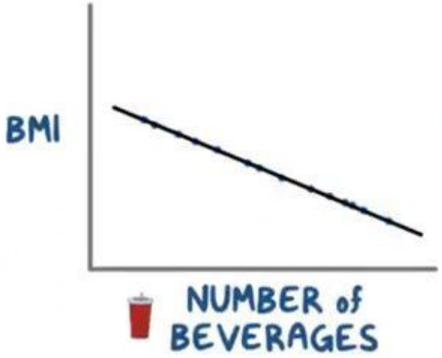
POSITIVE CORRELATION



PERFECT POSITIVE CORRELATION

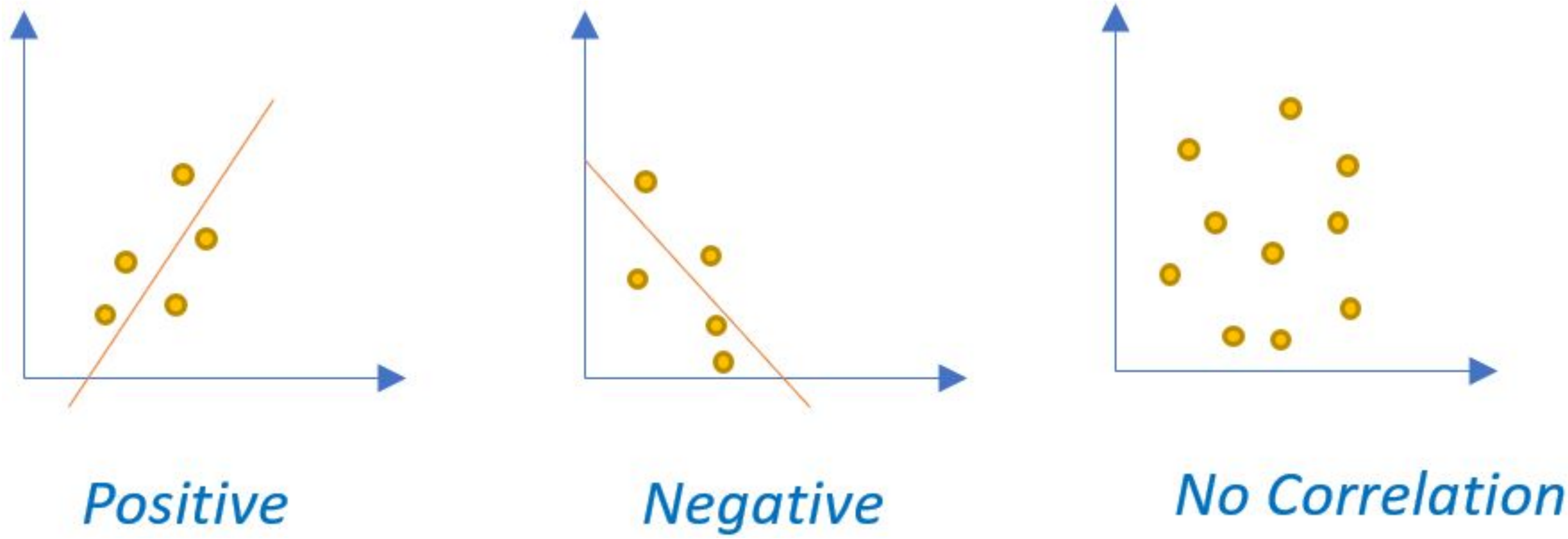


NEGATIVE CORRELATION

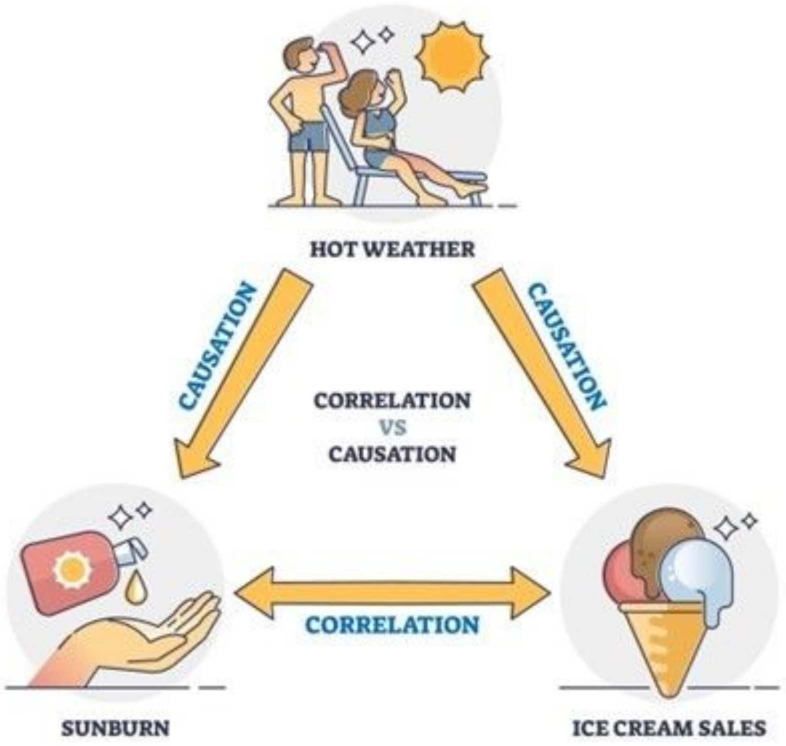


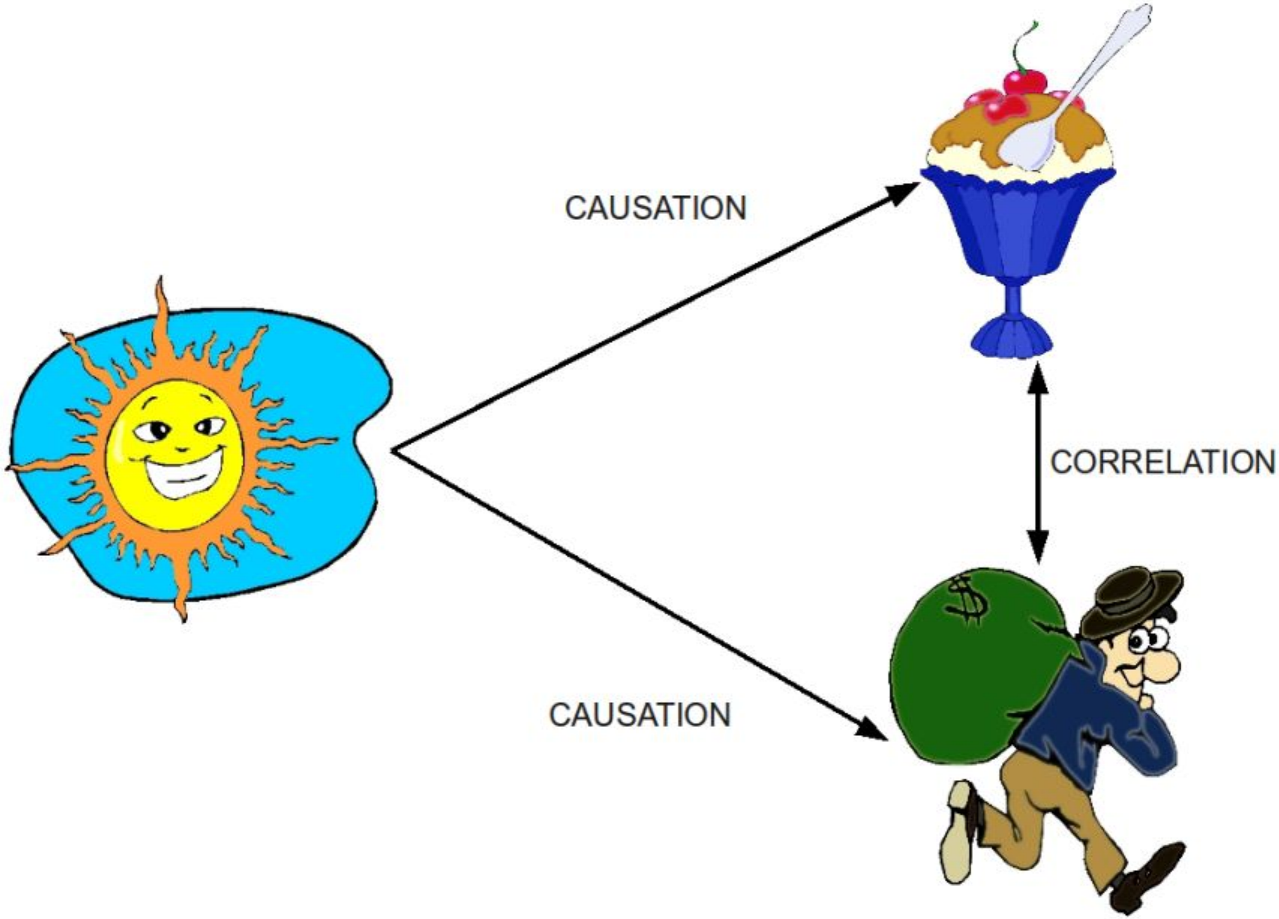
PERFECT NEGATIVE COP

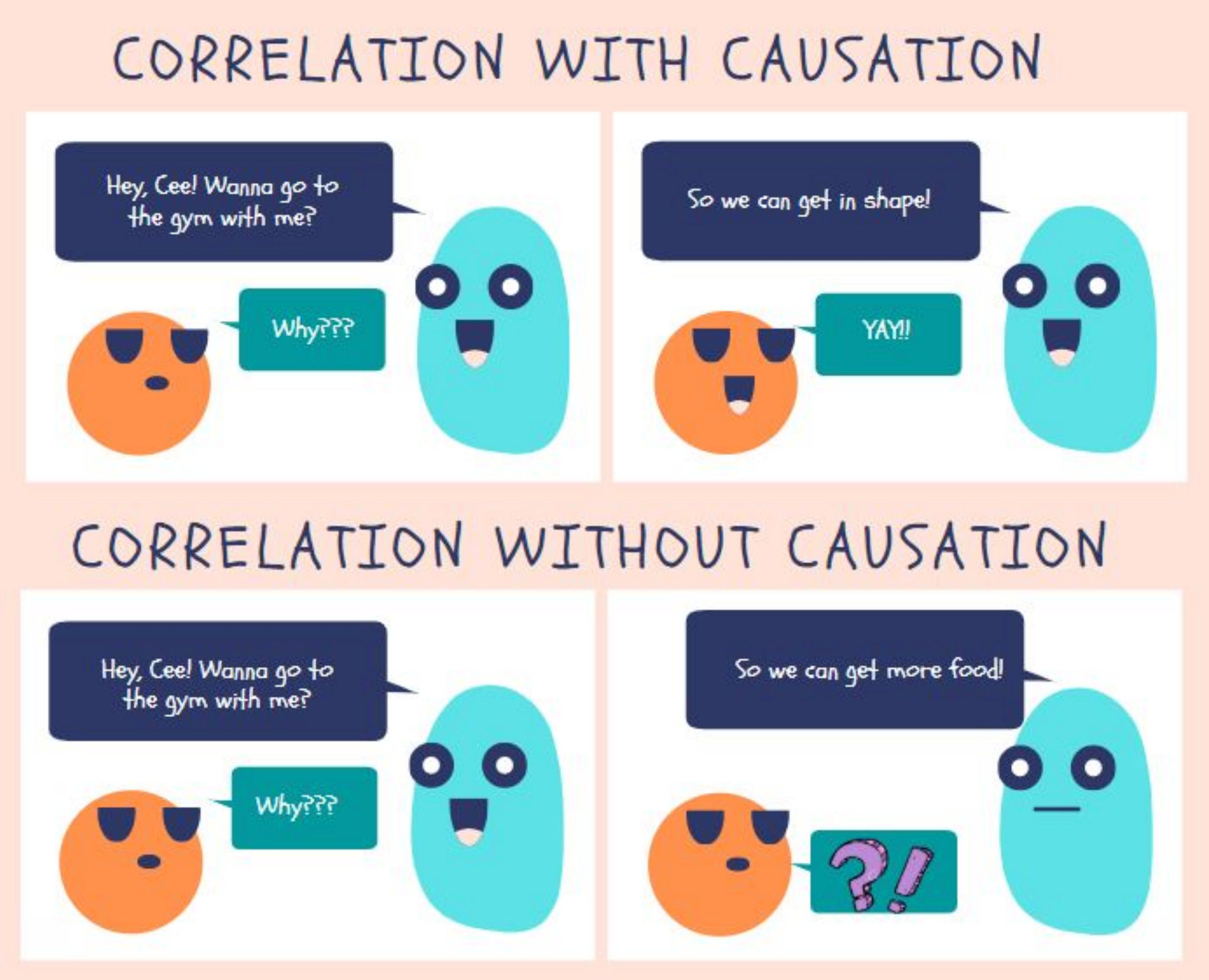
Types of Correlation Coefficients



Correlation Vs. Causality!




CAUSATION
CORRELATION
CAUSATION


CORRELATION WITH CAUSATION
Hey, Cee! Wanna go to the gym with me?
Why???
So we can get in shape!
YAY!!
CORRELATION WITHOUT CAUSATION
Hey, Cee! Wanna go to the gym with me?
Why???
So we can get more food!
?!

Pandas **get_dummies** (One-Hot Encoding)

Sample:

Pandas **get_dummies** (One-Hot Encoding)

City
Houston
Rome
Madrid
London

one-hot encoding

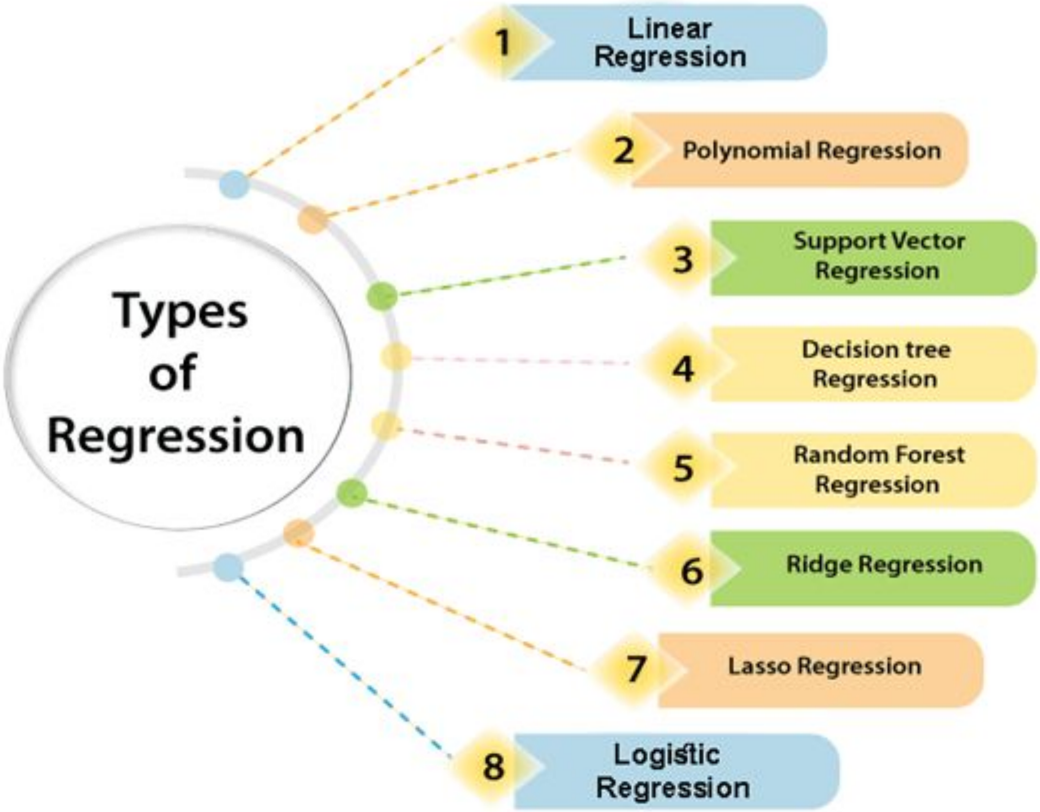
Houston	Rome	Madrid	London
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Minimum Mean Absolute Error (MAE)

$$MAE = \overset{\text{Divide by the total number of data points}}{\boxed{\frac{1}{n}}} \underset{\text{Sum of}}{\sum} \underbrace{\left| \overset{\text{Actual output value}}{\boxed{y}} - \overset{\text{Predicted output value}}{\boxed{\hat{y}}} \right|}_{\text{The absolute value of the residual}}$$

$$Error = Y_t - \hat{Y}_t$$

Mean squared error	$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$
Root mean squared error	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
Mean absolute error	$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$


Types of Regression
1
Linear Regression
2
Polynomial Regression
3
Support Vector Regression
4
Decision tree Regression
5
Random Forest Regression
6
Ridge Regression
7
Lasso Regression
8
Logistic Regression

Let's Start Coding: Regression Model

Import necessary libraries:

[2]

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
import statsmodels.api as sm
```

Load the dataset:

[3]

```
df = pd.read_csv('house_prices_dataset.csv')
```

EDA (Exploratory Data Analysis)

[4]

```
df.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl

5 rows × 81 columns

```
[5]  1 df.shape
```

```
(2919, 81)
```

```
[6]  1 df.columns
```

```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

Data Fields

- **SalePrice**: The property's sale price in dollars. This is the target variable that you're trying to predict.
- **MSSubClass**: The building class
- **MSZoning**: The general zoning classification
- **LotFrontage**: Linear feet of street connected to the property
- **LotArea**: Lot size in square feet
- **Street**: Type of road access
- **Alley**: Type of alley access
- **LotShape**: General shape of the property
- **LandContour**: Flatness of the property
- **Utilities**: Type of utilities available
- **LotConfig**: Lot configuration
- **LandSlope**: Slope of the property
- **Neighborhood**: Physical locations within Ames city limits
- **Condition1**: Proximity to the main road or railroad
- **Condition2**: Proximity to the main road or railroad (if a second is present)
- **BldgType**: Type of dwelling
- **HouseStyle**: Style of dwelling

Handle missing values:

```
# Check for missing values

missing_values = df.isnull().sum()
missing_values
```

```
Id                 0
MSSubClass         0
MSZoning           4
LotFrontage      486
LotArea            0
                ...
MoSold             0
YrSold             0
SaleType           1
SaleCondition      0
SalePrice       1459
Length: 81, dtype: int64
```

```python
missing_values = missing_values[missing_values > 0]
missing_values
```

```
MSZoning           4
LotFrontage      486
Alley           2721
Utilities          2
Exterior1st        1
Exterior2nd        1
MasVnrType        24
MasVnrArea        23
BsmtQual          81
BsmtCond          82
BsmtExposure      82
BsmtFinType1      79
BsmtFinSF1         1
BsmtFinType2      80
```

0s [9]

```python
# Replace missing values in numeric columns with mean and in categorical columns with mode

numeric_cols = df.select_dtypes(include=np.number).columns.tolist()
categorical_cols = df.select_dtypes(exclude=np.number).columns.tolist()

for col in numeric_cols:
    df[col].fillna(df[col].mean(), inplace=True)

for col in categorical_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

https://www.mashupmath.com/blog/mean-median-mode-range-guide

https://www.mashupmath.com/blog/mean-median-mode-range-guide

1 3 4 6 6 7 8

mean $= 35 \div 7 = 5$ The mean is 5 goals per game.

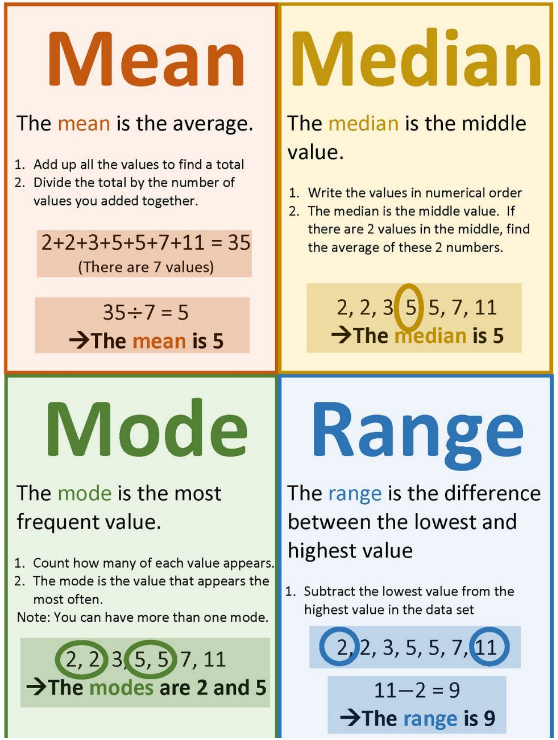
35: Total Sum; 7: Total Numbers

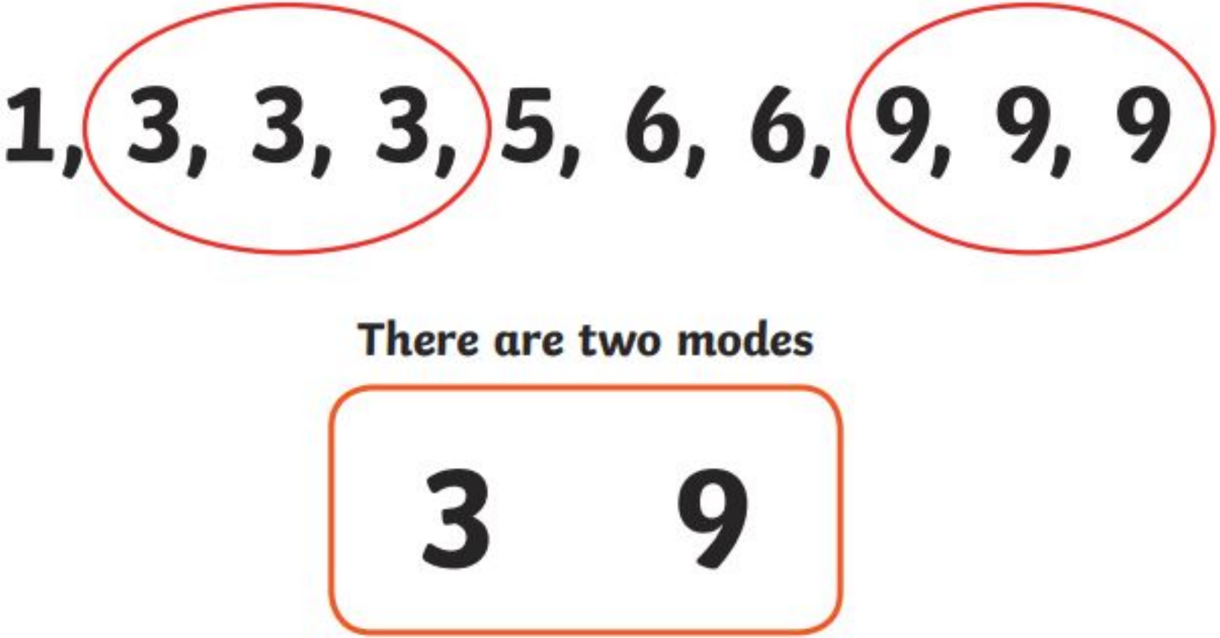
1 3 4 $\boxed{6 \quad 6}$ 7 8

mode $= 6$ The mode is 6 goals.

Look for numbers that repeat!

https://www.mashupmath.com/blog/mean-median-mode-range-guide





0s [10]

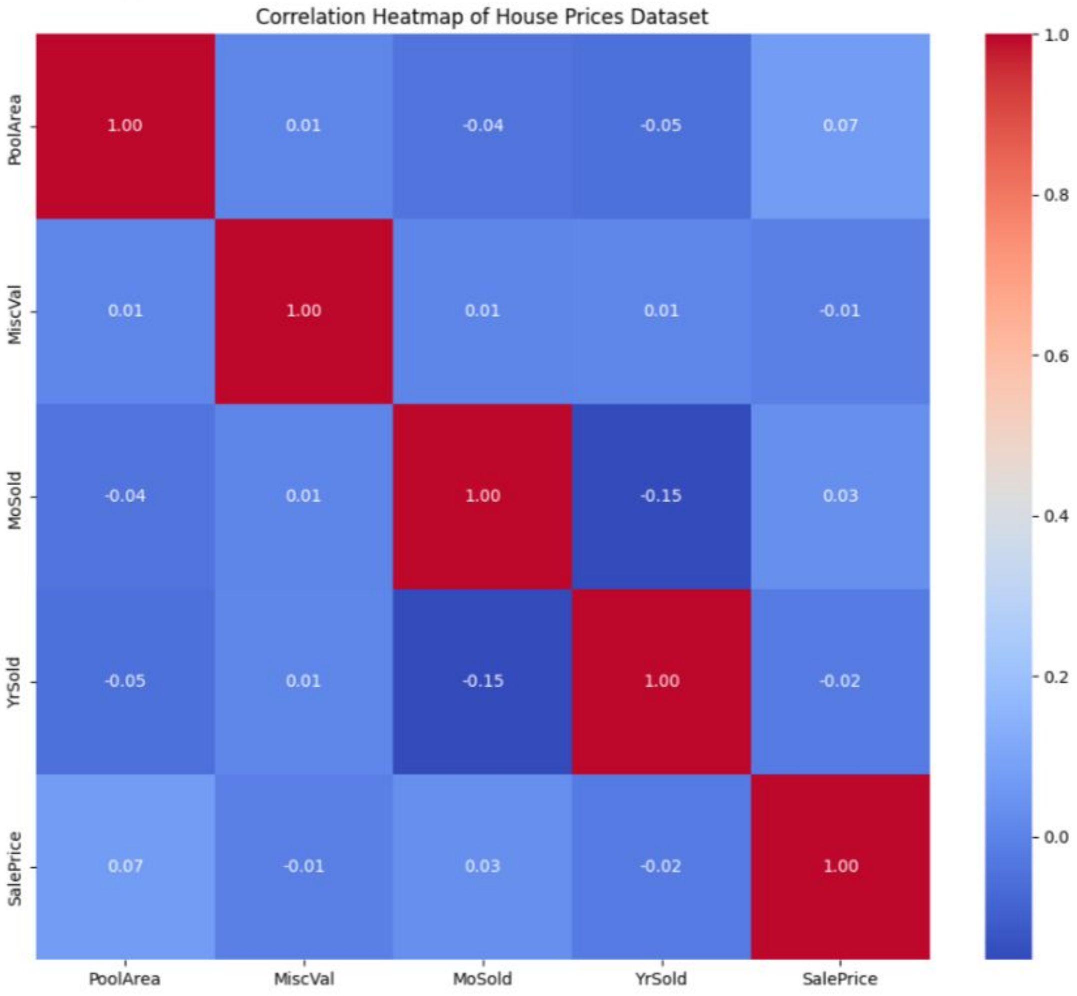
```python
# Reassessing for any remaining missing values

missing_values = df.isnull().sum()
missing_values
```

```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 81, dtype: int64
```

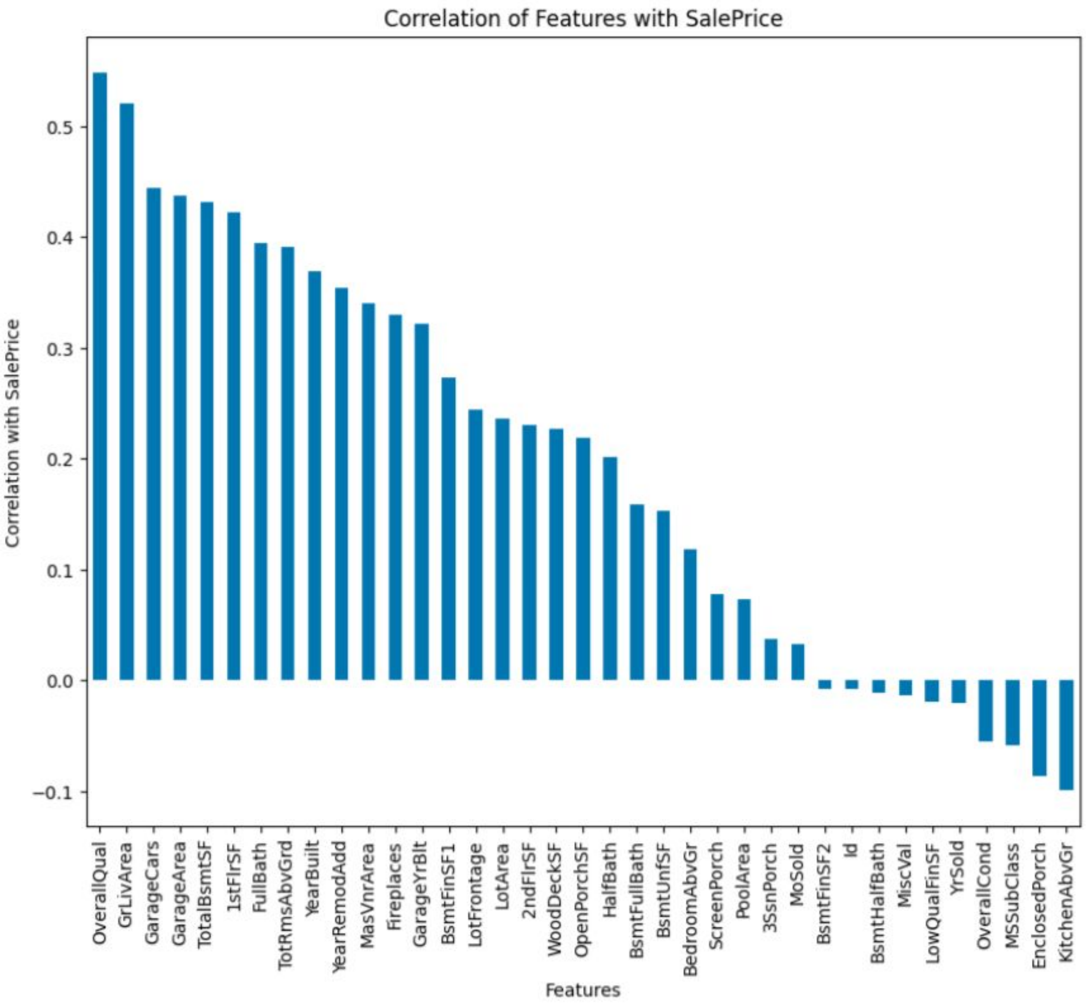
Correlation

- A statistic that measures the degree to which two securities move in relation to each other.

```python
# Calculate the correlations
correlation_matrix = df.iloc[:, -10:].corr()

# Set the figure size
plt.figure(figsize=(12, 10))

# Create a heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap of House Prices Dataset')
plt.show()
```


Correlation Heatmap of House Prices Dataset
PoolArea
MiscVal
MoSold
YrSold
SalePrice
1.00
0.01
-0.04
-0.05
0.07
0.01
1.00
0.01
0.01
-0.01
-0.04
0.01
1.00
-0.15
0.03
-0.05
0.01
-0.15
1.00
-0.02
0.07
-0.01
0.03
-0.02
1.00
PoolArea
MiscVal
MoSold
YrSold
SalePrice
1.0
0.8
0.6
0.4
0.2
0.0

```python
# Calculate correlations of features with the target 'SalePrice'
correlation_with_target = df.corr()['SalePrice'].sort_values(ascending=False)

# Plotting correlations with the target
plt.figure(figsize=(10, 8))
correlation_with_target.drop('SalePrice').plot(kind='bar')
plt.title('Correlation of Features with SalePrice')
plt.xlabel('Features')
plt.ylabel('Correlation with SalePrice')
plt.show()
```


Correlation of Features with SalePrice
Correlation with SalePrice
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
OverallQual
GrLivArea
GarageCars
GarageArea
TotalBsmtSF
1stFlrSF
FullBath
TotRmsAbvGrd
YearBuilt
YearRemodAdd
MasVnrArea
Fireplaces
GarageYrBlt
BsmtFinSF1
LotFrontage
LotArea
2ndFlrSF
WoodDeckSF
OpenPorchSF
HalfBath
BsmtFullBath
BsmtUnfSF
BedroomAbvGr
ScreenPorch
PoolArea
3SsnPorch
MoSold
BsmtFinSF2
Id
BsmtHalfBath
MiscVal
LowQualFinSF
YrSold
OverallCond
MSSubClass
EnclosedPorch
KitchenAbvGr
Features

```python
# Calculate correlations of features with the target 'SalePrice'
correlation_with_target = df.corr()['SalePrice'].abs().sort_values(ascending=False)

# Select the top five features correlated with SalePrice
top_five_features = correlation_with_target[1:6]  # Excluding SalePrice itself

# Display the top five correlated features
top_five_features
```

```
OverallQual     0.548617
GrLivArea       0.520311
GarageCars      0.444406
GarageArea      0.437654
TotalBsmtSF     0.431912
Name: SalePrice, dtype: float64
```

Dummy Code (One-Hot Encoding)

[14]

```
df['SaleCondition'].unique()
```

```
array(['Normal', 'Abnorml', 'Partial', 'AdjLand', 'Alloca', 'Family'],
      dtype=object)
```

[15]

```
# Assuming 'data' is your DataFrame containing both numerical and categorical columns
df = pd.get_dummies(df)
```

```
df.columns[-20:]
```

```
Index(['Fence_MnWw', 'MiscFeature_Gar2', 'MiscFeature_Othr',
       'MiscFeature_Shed', 'MiscFeature_TenC', 'SaleType_COD', 'SaleType_CWD',
       'SaleType_Con', 'SaleType_ConLD', 'SaleType_ConLI', 'SaleType_ConLw',
       'SaleType_New', 'SaleType_Oth', 'SaleType_WD', 'SaleCondition_Abnorml',
       'SaleCondition_AdjLand', 'SaleCondition_Alloca', 'SaleCondition_Family',
       'SaleCondition_Normal', 'SaleCondition_Partial'],
      dtype='object')
```

[14] 0s

```
df.shape # before using get_dummies
```

```
(2919, 81)
```

get_dummies

[18] 0s

```
df.shape # after using get_dummies
```

```
(2919, 290)
```

Feature selection based on correlation:

Replace 'Feature1', 'Feature2', 'Feature3', 'Feature4', 'Feature5' with the names of the top 5 features you've identified from the correlation plot.

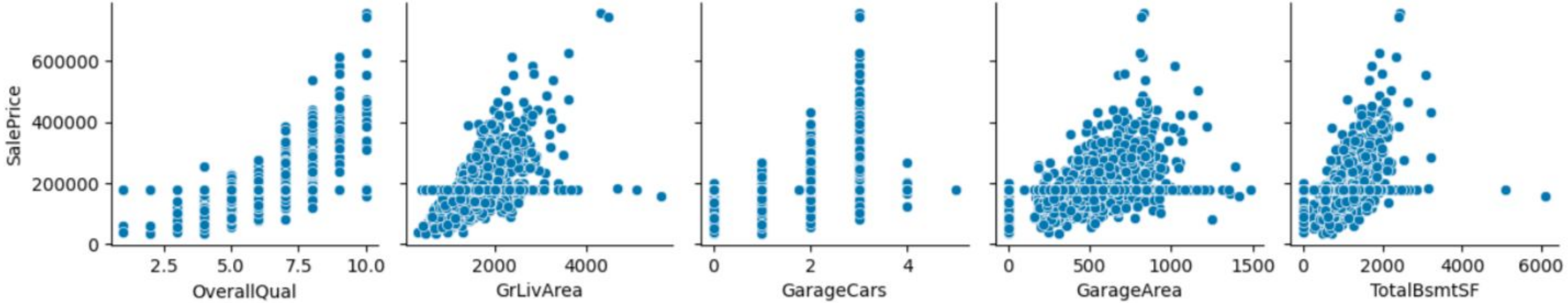
[14]
```
selected_features = ['Feature1', 'Feature2', 'Feature3', 'Feature4', 'Feature5']
```

[15]
```
selected_features = ['OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea', 'TotalBsmtSF']
```

[17]
```
X = df[selected_features]
y = df['SalePrice']
```

```python
1 sns.pairplot(data=df, y_vars=['SalePrice'], x_vars=selected_features)
```

```
<seaborn.axisgrid.PairGrid at 0x7f4640474250>
```



Split the dataset into training and testing sets:

[18]

```
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.2,
                                                    random_state=2024)
```

Create and fit a Linear Regression model:

[19]

```
model = LinearRegression()
```

[20]

```
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

[21]

```
y_pred = model.predict(X_test)
```

Calculate evaluation metrics: (RMSE, MAE, MSE)

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Print the calculated metrics
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
```

```
Mean Absolute Error (MAE): 31831.12330200309
Mean Squared Error (MSE): 1852270153.820561
Root Mean Squared Error (RMSE): 43038.00824643911
```

Perform statistical analysis using p-values and obtain regression model summary:

```python
X_train = sm.add_constant(X_train)
model_sm = sm.OLS(y_train, X_train).fit()
p_values = model_sm.pvalues
print("P-values:")
print(p_values)
print("Regression Model Summary:")
print(model_sm.summary())
```

```
P-values:
const          1.219841e-18
OverallQual    2.418788e-30
GrLivArea      1.007386e-38
GarageCars     2.729036e-01
GarageArea     4.382539e-02
TotalBsmtSF    1.102933e-10
dtype: float64
Regression Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.400
Model:                            OLS   Adj. R-squared:                  0.398
Method:                 Least Squares   F-statistic:                     310.0
Date:                Fri, 05 Jan 2024   Prob (F-statistic):          7.89e-255
Time:                        06:17:57   Log-Likelihood:                -28303.
No. Observations:                2335   AIC:                         5.662e+04
Df Residuals:                    2329   BIC:                         5.665e+04
Df Model:                           5
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const        3.672e+04   4131.783      8.888      0.000    2.86e+04    4.48e+04
OverallQual  1.092e+04    940.561     11.612      0.000    9077.150    1.28e+04
GrLivArea      30.2332      2.281     13.257      0.000      25.761      34.705
GarageCars   2996.5169   2732.404      1.097      0.273   -2361.680    8354.714
GarageArea     19.3333      9.586      2.017      0.044       0.536      38.131
TotalBsmtSF    16.9178      2.610      6.482      0.000      11.799      22.036
==============================================================================
Omnibus:                     1075.456   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            16166.517
Skew:                           1.780   Prob(JB):                         0.00
Kurtosis:                      15.389   Cond. No.                     8.99e+03
==============================================================================
```

```
P-values:
const          1.219841e-18
OverallQual    2.418788e-30
GrLivArea      1.007386e-38
GarageCars     2.729036e-01
GarageArea     4.382539e-02
TotalBsmtSF    1.102933e-10
dtype: float64
Regression Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.400
Model:                            OLS   Adj. R-squared:                  0.398
Method:                 Least Squares   F-statistic:                     310.0
Date:                Fri, 05 Jan 2024   Prob (F-statistic):          7.89e-255
Time:                        06:17:57   Log-Likelihood:                -28303.
No. Observations:                2335   AIC:                         5.662e+04
Df Residuals:                    2329   BIC:                         5.665e+04
Df Model:                           5
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const        3.672e+04   4131.783      8.888      0.000    2.86e+04    4.48e+04
OverallQual  1.092e+04    940.561     11.612      0.000    9077.150    1.28e+04
GrLivArea      30.2332      2.281     13.257      0.000      25.761      34.705
GarageCars   2996.5169   2732.404      1.097      0.273   -2361.680    8354.714
GarageArea     19.3333      9.586      2.017      0.044       0.536      38.131
TotalBsmtSF    16.9178      2.610      6.482      0.000      11.799      22.036
==============================================================================
Omnibus:                     1075.456   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            16166.517
Skew:                           1.780   Prob(JB):                         0.00
Kurtosis:                      15.389   Cond. No.                     8.99e+03
==============================================================================
```

- **Explain the regression equation**

Week 7: Assignment

Apply **regression modelling** techniques to **predict house prices** accurately.

The primary objective of this assignment is to create regression models using the House Prices dataset to predict property sale prices accurately.