When to use linear regression

1: Your two variables should be measured at the **continuous** level.
2: There needs to be a **linear relationship** between the two variables.
3: There should be **no significant outliers**.
4: You should have **independence of observations**,
5: Your data needs to show **homoscedasticity**

Dataset

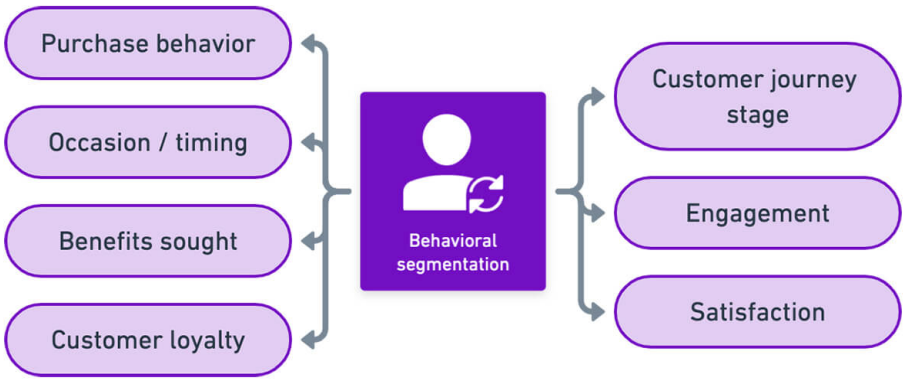
Variable	Type	Description
Number_claims	Discrete	Number of claims received from the insured per year
Industrial_city	Binary	Takes the value 1 if insured lives in industrial city (Casablanca, Mohammedia, Kenitra or Tanger), 0 otherwise.
Gender_male	Binary	Takes the value 1 if the insured is male, 0 if the inured is female
Industrial_activity	Binary	Takes the value 1 if insured works in industrial firm, 0 otherwise
Services_activity	Binary	Takes the value 1 if the insured works in the services company (e.g insurance and bank), 0 otherwise
Age_30	Binary	Takes the value 1 if the insured have an age less than 30 years, 0 otherwise
Age_40	Binary	Takes the value 1 if the age of insured varies between 30 and 40 years, 0 otherwise
Age_60	Binary	Takes the value 1 if the age of insured varies between 40 and 60 years, 0 otherwise
Status_married	Binary	Takes the value 1 if insured is married, 0 otherwise
Status_single	Binary	Takes the value 1 if insured is single, 0 otherwise
Size_family	Discrete	Indicates the size of the family of the insured person
Exposure	Continuous	Indicates coverage period of the insured in the year. It varies between 0 and 1

	age	bmi	children	charges	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	19	27.900	0	16884.92400	0	1	0	0	1
1	18	33.770	1	1725.55230	1	0	0	1	0
2	28	33.000	3	4449.46200	1	0	0	1	0
3	33	22.705	0	21984.47061	1	0	1	0	0
4	32	28.880	0	3866.85520	1	0	1	0	0

When to use Logistic regression

1) When two Class prediction problem like (0/1 or male/female or yes/no)
2) Dataset with High Variance and Low Bias
3) NO Significant Outliers
4) Required Large Dataset
5) Absence of multicollinearity
6) Remove correlated inputs

Dataset
customerID object
gender object
SeniorCitizen int64
Partner object
Dependents object
tenure int64
PhoneService object
MultipleLines object
InternetService object
OnlineSecurity object
OnlineBackup object
DeviceProtection object
TechSupport object
StreamingTV object
StreamingMovies object
Contract object
PaperlessBilling object
PaymentMethod object
MonthlyCharges float64
TotalCharges object
Churn object



When to use Kmeans Clustering

1. Unlabeled Data Sets.
2. Nonlinearly Separable Data.
3. Speed.
4. K-Means Clustering is a simple yet powerful algorithm in data science.
5. Scales to large data sets.
6. Easily adapts to new examples.
7. Guarantees convergence.
8. Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

class sklearn.cluster.KMeans(*n_clusters=8*, *, *init='k-means++'*, *n_init=10*, *max_iter=300*, *tol=0.0001*, *precompute_distances='deprecated'*, *verbose=0*, *random_state=None*, *copy_x=True*, *n_jobs='deprecated'*, *algorithm='auto'*)

Interesting use cases for k-means clustering in business

- Consumer segmentation
- Delivery optimisation
- Document sorting and grouping
- Customer retention
- Discount analysis

Dataset:

Spending Score (1-100)
Score assigned by the mall based on customer behavior and spending nature

CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39

InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with the letter 'c', it indicates a cancellation.
StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description: Product (item) name. Nominal.
Quantity: The quantities of each product (item) per transaction. Numeric.
InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice: Unit price. Numeric, Product price per unit in sterling.
CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country: Country name. Nominal, the name of the country where each customer resides.

When to use Random Forest Algorithm

1. Random forest algorithm can be used for both classifications and regression task.
2. It provides higher accuracy through cross validation.
3. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.
4. If there are more trees, it won't allow over-fitting trees in the model.
5. It has the power to handle a large data set with higher dimensionality

class sklearn.ensemble.RandomForestClassifier(*n_estimators=100*, *, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *bootstrap=True*, *oob_score=False*, *n_jobs=None*, *random_state=None*, *verbose=0*, *warm_start=False*, *class_weight=None*, *ccp_alpha=0.0*, *max_samples=None*)

Dataset
customerID object
gender object
SeniorCitizen int64
Partner object
Dependents object
tenure int64
PhoneService object
MultipleLines object
InternetService object
OnlineSecurity object
OnlineBackup object
DeviceProtection object
TechSupport object
StreamingTV object
StreamingMovies object
Contract object
PaperlessBilling object
PaymentMethod object
MonthlyCharges float64
TotalCharges object
Churn object