# *A Guide for Data Science Analytics*

*Data Cleaning accounts for 80% of the work in industry. It is not simply 'garbage in, garbage out' [1], it is an important part of the pre-processing pipeline. Proper data cleaning (feature engineering, data visualisation and encoding), ensures quality and accuracy of the data used in analytics and machine learning (ML) tools.*

## Analytic tools



## *Introduction*

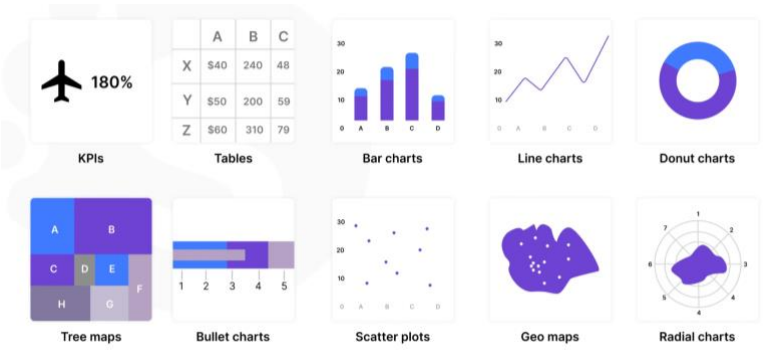
Examining datasets and using measurements to understand variables will allow you to find intricacies in the data '...to gain insights, refine features, and prepare the data for model training' [2]. Before using tools for analyses, you need to understand what kind of data you have, '90% of all the data is unstructured' [3]. You need to have the right methods to explore, visualise and find statistics to measure your data before applying Machine learning tools.

Key components used in data analysis include,

***Feature Engineering*** *to transform and create new features to enhance predictive elements of a Machine Learning (ML) model. This is also known as 'Data Exploration', the initial examination of data to find patterns, anomalies, and insights [2].*

***Data Visualisation*** *Graphical representation of data to identify trends and relationships. Using data to understand patterns, outliers and distribution in the dataset, to identify trends, potential issues and correlations, this helps to support informed decisions when selecting primary keys for predictions and developing models.*

***Interval Scale*** *converts categorical variables into a format a model can understand (numerical), this enables easy use of ML algorithms and interpretation from all data types in the dataset. This is also known as 'Descriptive statistics', Summary statistics that describe the basic features of the data. Example: Mean, Median, Model [1].*




*Different Data Visualisations [3]*

## Types of Measurement Scales

Measurement scales are fundamental in data science, they determine the nature of the data and the types of analyses that can be performed.

***Nominal Scale*** *data without a specific order. i.e. Gender, Eye Colour.*

***Ordinal Scale*** *data with a specific order but without a standard interval between categories. i.e. Survey Ratings (Poor, Fair, Good).*

***Interval Scale*** *data with meaningful intervals but no true zero point. i.e. Temperature in Celsius.*

***Ratio Scale*** *data with meaningful intervals and a true zero point. i.e. Weight, Height.*

## Importing Libraries

These libraries create a great set of tools for data visualisation, manipulation and machine learning for analysis. Each can be combined to enable navigation of data structure, visualise insights and utilise ML algorithm tools.

```
[ ]  1 import pandas as pd
     2 import numpy as np
     3 import matplotlib.pyplot as plt
     4 import seaborn as sns
     5 from sklearn.model_selection import train_test_split
     6 from sklearn.ensemble import RandomForestRegressor
```

Using 'pd.read_csv' we can transform raw data into a python Data Frame for analysis.

```
1 train_data = pd.read_csv('train.cs')
```

Train datasets

```
3 test_data = pd.read_csv('test.csv')
4
```

Test datasets

## Analytics

Analytics involves the systematic computational analysis of data. Key steps include:

**Data Collection** gathering data from various sources.

**Data Cleaning** removing or correcting errors and inconsistencies in the data.

**Data Transformation** converting data into a suitable format for analysis.

**Exploratory Data Analysis (EDA)** analysing data sets to summarize their main characteristics.

Variables can be classified into,

***Independent Variables*** *influence or predict the outcome.*

***Dependent Variables*** *predicted by the independent variables.*

***Continuous Variables*** *take any value within a range. i.e. Age, Salary.*

***Categorical Variables*** *a limited number of values, representing different categories. i.e. Type of Car.*

```
1 # pedestrian-counting-system-monthly-counts-per-hour
2 print(datasets['pedestrian-counting-system-monthly-counts-per-hour'].head())
```

```
   sensor_name                 timestamp  locationid  direction_1  \
0    SprFli_T  2023-04-24T21:00:00+00:00          75           36
1    SprFli_T  2023-04-25T00:00:00+00:00          75           28
2    SprFli_T  2023-04-25T01:00:00+00:00          75           63
3    SprFli_T  2023-04-25T02:00:00+00:00          75           85
4    SprFli_T  2023-04-25T08:00:00+00:00          75          365

   direction_2  total_of_directions                  location
0           17                   53  -37.81515276, 144.97467661
1           50                   78  -37.81515276, 144.97467661
2           63                  126  -37.81515276, 144.97467661
3           89                  174  -37.81515276, 144.97467661
4           59                  424  -37.81515276, 144.97467661
```

Before looking at intricacies of the dataset, peer at the first few rows of the dataset 'df.head()', it provides a snapshot of the data's structure and content.

```
1 ped_count_hour.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 549976 entries, 0 to 549975
Data columns (total 7 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   sensor_name          549976 non-null  object
 1   timestamp            549976 non-null  datetime64[ns, UTC]
 2   locationid           549976 non-null  int64
 3   direction_1          549976 non-null  int64
 4   direction_2          549976 non-null  int64
 5   total_of_directions  549976 non-null  int64
 6   location             549976 non-null  object
dtypes: datetime64[ns, UTC](1), int64(4), object(2)
memory usage: 29.4+ MB
```

```
1 ped_count_hour.describe()
```

| | locationid | direction_1 | direction_2 | total_of_directions |
|---|---|---|---|---|
| count | 549976.000000 | 549976.000000 | 549976.000000 | 549976.000000 |
| mean | 53.513222 | 211.134159 | 211.066339 | 422.200498 |
| std | 36.208624 | 324.106316 | 321.837839 | 615.577558 |
| min | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 25% | 24.000000 | 19.000000 | 20.000000 | 42.000000 |
| 50% | 51.000000 | 86.000000 | 84.000000 | 177.000000 |
| 75% | 72.000000 | 257.000000 | 255.000000 | 530.000000 |
| max | 142.000000 | 8900.000000 | 8089.000000 | 10387.000000 |

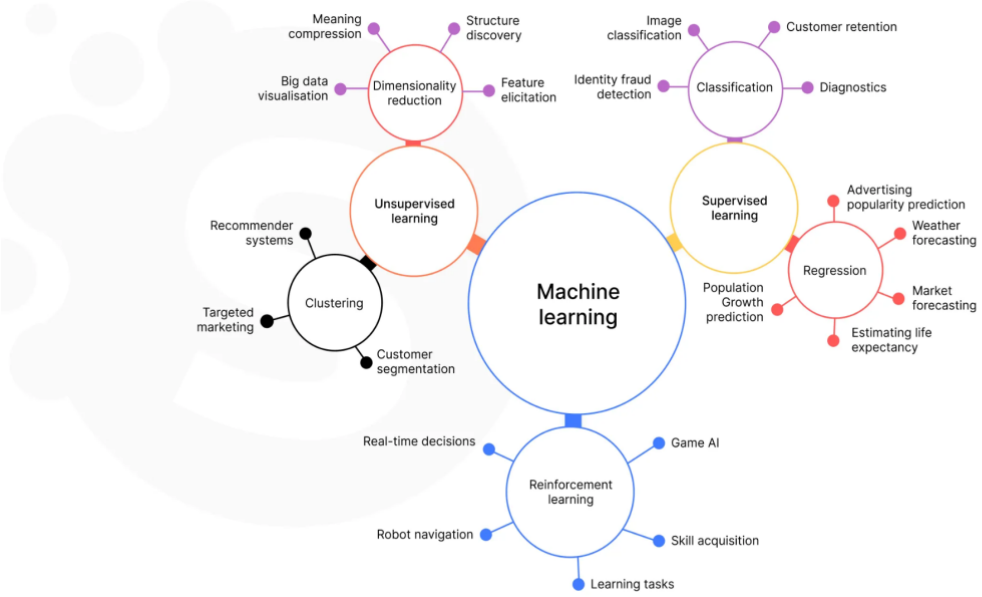
Method prints out information about the Data Frame df.head() , this information contains number of columns, labels, data types, memory usage, range index and number of non-null values.

Method df.describe() , is used for calculating statistical properties from the DataFrame, percentile, mean, std (it will analyse both numerical and object series in a DataFrame - including mixed data types!).

## Machine Learning

Machine Learning (ML) tools are used to build models to make predictions based on the data you have collected [3], building systems that can learn from data, identify patterns, and make decisions with minimal manual input.

Types of machine learning,




**Supervised Learning** model is trained on a labeled dataset, which means that each training example is paired with an output label, the model learns a mapping from inputs to find the correct output. Like assigning labels to new instances (i.e. spam detection in emails - classification) or predicting a continuous value (ie. predicting house prices - regression).

**Unsupervised Learning** training a model on data without labeled responses, to find hidden patterns or structures in the input data. Grouping similar instances together ie. Clustering or reducing the number of variables under consideration ie. principal component analysis - dimensionality reduction.

**Reinforcement Learning** the model learns to make decisions by taking actions in an environment to maximise reward (weights), learns from consequences of actions not from a labeled dataset.

**Transfer Learning** takes a pre-trained model and adapts it to a different task, used when there is limited data for the new task!

**Federated Learning** a decentralised method - models are trained across multiple servers with local data samples, without sharing them, good for data privacy and security!

**Active Learning** a model can interactively query a user/ database to find new outputs with new data points - good when labels are hard to identify.

Libraries for ML include,

**Scikit-learn** a Python library for simple and efficient tools for data mining and data analysis. https://scikit-learn.org/

**TensorFlow** open-source library for numerical computation and large-scale machine learning. https://www.tensorflow.org/

**Keras** neural networks API, written in Python and capable of running on top of TensorFlow. https://keras.io/

**PyTorch** machine learning library based on the Torch library. https://pytorch.org/

**Hugging Face** a platform to build, deploy and train machine learning models, datasets and applications. https://huggingface.co/

## Further reading

[1] Fokoye, "UNDERSTANDING YOUR DATA: BEGINNER'S GUIDE TO DATA ANALYTICS/DATA SCIENCE," *Medium*, Nov. 22, 2023. [Online]. Available: https://medium.com/@fokoye/understanding-your-data-beginners-guide-to-data-analytics-data-science-1826d4d82e2f

[2] U. Fahad, "A comprehensive Guide to Data Analysis in Machine Learning: Fundamental and Techniques", Medium, Jan. 24, 2024. [Online]. https://medium.com/@rokhaiyasultana97/a-comprehensive-guide-to-data-analysis-in-machine-learning-fundamentals-and-techniques-3f3b3ce2c9d3

[3] S. Vaniukov, "Advanced Data Analytics: A Guide to Practical Applications, Medium, OCt. 21, 2023. [Online]. Available: https://medium.com/@vaniukov.s/advanced-data-analytics-a-guide-to-practical-applications-a47c8807a85f

## Author

Te' Claire 2024.v1