Experiment 1

In [1]:
```
import pandas as pd
```

In [2]:
```
my_data = pd.read_csv("http://winterolympicsmedals.com/medals.csv")
```

In [3]:
```
my_data
```

Out[3]:

	Year	City	Sport	Discipline	NOC	Event	Event gender	Medal
0	1924	Chamonix	Skating	Figure skating	AUT	individual	M	Silver
1	1924	Chamonix	Skating	Figure skating	AUT	individual	W	Gold
2	1924	Chamonix	Skating	Figure skating	AUT	pairs	X	Gold
3	1924	Chamonix	Bobsleigh	Bobsleigh	BEL	four-man	M	Bronze
4	1924	Chamonix	Ice Hockey	Ice Hockey	CAN	ice hockey	M	Gold
...	...	...	...	...	...	...	...	...
2306	2006	Turin	Skiing	Snowboard	USA	Half-pipe	M	Silver
2307	2006	Turin	Skiing	Snowboard	USA	Half-pipe	W	Gold
2308	2006	Turin	Skiing	Snowboard	USA	Half-pipe	W	Silver
2309	2006	Turin	Skiing	Snowboard	USA	Snowboard Cross	M	Gold
2310	2006	Turin	Skiing	Snowboard	USA	Snowboard Cross	W	Silver

2311 rows × 8 columns

In [4]:
```
pip install opendatasets
```

```
Requirement already satisfied: opendatasets in /Users/chiragchan/opt/anaconda3/envs/data
_visualisation_python/lib/python3.9/site-packages (0.1.22)
Requirement already satisfied: tqdm in /Users/chiragchan/opt/anaconda3/envs/data_visuali
sation_python/lib/python3.9/site-packages (from opendatasets) (4.64.1)
Requirement already satisfied: click in /Users/chiragchan/opt/anaconda3/envs/data_visual
isation_python/lib/python3.9/site-packages (from opendatasets) (8.1.3)
Requirement already satisfied: kaggle in /Users/chiragchan/opt/anaconda3/envs/data_visua
lisation_python/lib/python3.9/site-packages (from opendatasets) (1.5.12)
Requirement already satisfied: python-slugify in /Users/chiragchan/opt/anaconda3/envs/da
ta_visualisation_python/lib/python3.9/site-packages (from kaggle->opendatasets) (6.1.2)
Requirement already satisfied: certifi in /Users/chiragchan/opt/anaconda3/envs/data_visu
alisation_python/lib/python3.9/site-packages (from kaggle->opendatasets) (2022.9.24)
Requirement already satisfied: python-dateutil in /Users/chiragchan/opt/anaconda3/envs/d
ata_visualisation_python/lib/python3.9/site-packages (from kaggle->opendatasets) (2.8.2)
Requirement already satisfied: urllib3 in /Users/chiragchan/opt/anaconda3/envs/data_visu
alisation_python/lib/python3.9/site-packages (from kaggle->opendatasets) (1.26.11)
Requirement already satisfied: requests in /Users/chiragchan/opt/anaconda3/envs/data_vis
ualisation_python/lib/python3.9/site-packages (from kaggle->opendatasets) (2.28.1)
Requirement already satisfied: six>=1.10 in /Users/chiragchan/opt/anaconda3/envs/data_vi
sualisation_python/lib/python3.9/site-packages (from kaggle->opendatasets) (1.16.0)
Requirement already satisfied: text-unidecode>=1.3 in /Users/chiragchan/opt/anaconda3/en
vs/data_visualisation_python/lib/python3.9/site-packages (from python-slugify->kaggle->o
pendatasets) (1.3)
Requirement already satisfied: idna<4,>=2.5 in /Users/chiragchan/opt/anaconda3/envs/data
_visualisation_python/lib/python3.9/site-packages (from requests->kaggle->opendatasets)
(3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in /Users/chiragchan/opt/anacond
```

```
a3/envs/data_visualisation_python/lib/python3.9/site-packages (from requests->kaggle->op
endatasets) (2.1.1)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
#Import directly from repositories using URL
import opendatasets as od
dataset_url = 'https://www.kaggle.com/datasets/ruchi798/,data-science-job-salaries/downl
od.download('https://www.kaggle.com/datasets/ruchi798/data-science-job-salaries/download
```

```
Skipping, found downloaded files in "./data-science-job-salaries" (use force=True to for
ce download)
```

In [6]:
```
#Import dataset from a csv
data = pd.read_csv("data-science-job-salaries/ds_salaries.csv")
```

In [7]:
```
data.head()
```

Out[7]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_ir
0	0	2020	MI	FT	Data Scientist	70000	EUR	7
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	26
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	1C
3	3	2020	MI	FT	Product Data Analyst	20000	USD	2
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	15

In [8]:
```
pip install openpyxl
```

```
Requirement already satisfied: openpyxl in /Users/chiragchan/opt/anaconda3/envs/data_vis
ualisation_python/lib/python3.9/site-packages (3.0.10)
Requirement already satisfied: et-xmlfile in /Users/chiragchan/opt/anaconda3/envs/data_v
isualisation_python/lib/python3.9/site-packages (from openpyxl) (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [9]:
```
import os
dir = os.getcwd()
filename = dir + "/../assets/xlsx/tips.xlsx"
print(filename)
```

```
/Users/chiragchan/Desktop/DV Programs/src/../assets/xlsx/tips.xlsx
```

In [10]:
```
mydata = pd.read_excel(filename)
```

In [11]:
```
from sklearn import datasets
```

In [12]:
```
dataset = datasets.load_breast_cancer()
```

In [13]:
```
dataset
```

Out[13]:
```
{'data': array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,
         1.189e-01],
        [2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,
         8.902e-02],
        [1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,
```

```
         8.758e-02],
        ...,
        [1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,
         7.820e-02],
        [2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01, 4.087e-01,
         1.240e-01],
        [7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,
         7.039e-02]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
        1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
        1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
        1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
        0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,
        1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
        0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
        1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
        1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,
        0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
        0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
        1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
        1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,
        1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
        1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
        1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1,
        1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1]),
 'frame': None,
 'target_names': array(['malignant', 'benign'], dtype='<U9'),
 'DESCR': '.. _breast_cancer_dataset:\n\nBreast cancer wisconsin (diagnostic) dataset\n-
--------------------------------------------\n\n**Data Set Characteristics:**\n\n    :Num
ber of Instances: 569\n\n    :Number of Attributes: 30 numeric, predictive attributes an
d the class\n\n    :Attribute Information:\n        - radius (mean of distances from cen
ter to points on the perimeter)\n        - texture (standard deviation of gray-scale val
ues)\n        - perimeter\n        - area\n        - smoothness (local variation in radi
us lengths)\n        - compactness (perimeter^2 / area - 1.0)\n        - concavity (seve
rity of concave portions of the contour)\n        - concave points (number of concave po
rtions of the contour)\n        - symmetry\n        - fractal dimension ("coastline appr
oximation" - 1)\n\n        The mean, standard error, and "worst" or largest (mean of the
three\n        worst/largest values) of these features were computed for each image,\n
      resulting in 30 features.  For instance, field 0 is Mean Radius, field\n        10
is Radius SE, field 20 is Worst Radius.\n\n        - class:\n                - WDBC-Mali
gnant\n                - WDBC-Benign\n\n    :Summary Statistics:\n\n    ================
==================== ====== ======\n                                                  Min    M
ax\n    ===================================== ====== ======\n    radius (mean):
              6.981  28.11\n    texture (mean):                        9.71   39.28\n
perimeter (mean):                     43.79  188.5\n    area (mean):
      143.5  2501.0\n    smoothness (mean):                     0.053  0.163\n    compact
ness (mean):                   0.019  0.345\n    concavity (mean):
0.0    0.427\n    concave points (mean):                0.0    0.201\n    symmetry (mea
n):                      0.106  0.304\n    fractal dimension (mean):              0.05
0.097\n    radius (standard error):              0.112  2.873\n    texture (standard err
or):             0.36   4.885\n    perimeter (standard error):           0.757  21.98\n
   area (standard error):                6.802  542.2\n    smoothness (standard error):
       0.002  0.031\n    compactness (standard error):         0.002  0.135\n    conca
vity (standard error):           0.0    0.396\n    concave points (standard error):
0.0    0.053\n    symmetry (standard error):            0.008  0.079\n    fractal dimens
ion (standard error):   0.001  0.03\n    radius (worst):                         7.93   3
6.04\n    texture (worst):                      12.02  49.54\n    perimeter (worst):
```

```
                50.41  251.2\n    area (worst):                          185.2  4254.0\n
   smoothness (worst):                   0.071  0.223\n    compactness (worst):
         0.027  1.058\n    concavity (worst):                     0.0    1.252\n    conca
ve points (worst):               0.0    0.291\n    symmetry (worst):
0.156  0.664\n    fractal dimension (worst):            0.055  0.208\n    ==============
======================== ====== ======\n\n    :Missing Attribute Values: None\n\n    :Cla
ss Distribution: 212 - Malignant, 357 - Benign\n\n    :Creator:  Dr. William H. Wolberg,
W. Nick Street, Olvi L. Mangasarian\n\n    :Donor: Nick Street\n\n    :Date: November, 1
995\n\nThis is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.\nhttps://
goo.gl/U2Uwz2\n\nFeatures are computed from a digitized image of a fine needle\naspirate
(FNA) of a breast mass.  They describe\ncharacteristics of the cell nuclei present in th
e image.\n\nSeparating plane described above was obtained using\nMultisurface Method-Tre
e (MSM-T) [K. P. Bennett, "Decision Tree\nConstruction Via Linear Programming." Proceedi
ngs of the 4th\nMidwest Artificial Intelligence and Cognitive Science Society,\npp. 97-1
01, 1992], a classification method which uses linear\nprogramming to construct a decisio
n tree.  Relevant features\nwere selected using an exhaustive search in the space of 1-4
\nfeatures and 1-3 separating planes.\n\nThe actual linear program used to obtain the se
parating plane\nin the 3-dimensional space is that described in:\n[K. P. Bennett and O.
L. Mangasarian: "Robust Linear\nProgramming Discrimination of Two Linearly Inseparable S
ets",\nOptimization Methods and Software 1, 1992, 23-34].\n\nThis database is also avail
able through the UW CS ftp server:\n\nftp ftp.cs.wisc.edu\ncd math-prog/cpo-dataset/mach
ine-learn/WDBC/\n\n.. topic:: References\n\n   - W.N. Street, W.H. Wolberg and O.L. Mang
asarian. Nuclear feature extraction \n     for breast tumor diagnosis. IS&T/SPIE 1993 In
ternational Symposium on \n     Electronic Imaging: Science and Technology, volume 1905,
pages 861-870,\n     San Jose, CA, 1993.\n   - O.L. Mangasarian, W.N. Street and W.H. Wo
lberg. Breast cancer diagnosis and \n     prognosis via linear programming. Operations R
esearch, 43(4), pages 570-577, \n     July-August 1995.\n   - W.H. Wolberg, W.N. Street,
and O.L. Mangasarian. Machine learning techniques\n     to diagnose breast cancer from f
ine-needle aspirates. Cancer Letters 77 (1994) \n     163-171.',
 'feature_names': array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
        'mean smoothness', 'mean compactness', 'mean concavity',
        'mean concave points', 'mean symmetry', 'mean fractal dimension',
        'radius error', 'texture error', 'perimeter error', 'area error',
        'smoothness error', 'compactness error', 'concavity error',
        'concave points error', 'symmetry error',
        'fractal dimension error', 'worst radius', 'worst texture',
        'worst perimeter', 'worst area', 'worst smoothness',
        'worst compactness', 'worst concavity', 'worst concave points',
        'worst symmetry', 'worst fractal dimension'], dtype='<U23'),
 'filename': 'breast_cancer.csv',
 'data_module': 'sklearn.datasets.data'}
```

In [14]:
```
import pandas as pd
df = pd.DataFrame(dataset.data, columns=dataset.feature_names)
```

In [15]:
```
df.head()
```

Out[15]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mea fract dimensic
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.078
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.056
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.059
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.097
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.058

5 rows × 30 columns

In [16]:
```
from keras.datasets import mnist
import matplotlib as plt
```

```
import numpy as np
```

In [17]:
```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

In [18]:
```
print(np.shape(x_train))
```

```
(60000, 28, 28)
```

In [19]:
```
np.shape(y_train)
```

Out[19]:
```
(60000,)
```

In [20]:
```
np.shape(x_test)
```

Out[20]:
```
(10000, 28, 28)
```

In [21]:
```
np.shape(y_test)
```

Out[21]:
```
(10000,)
```