PRABHAT DHAR

REG NO 2108152

TITLE

Design and Application of a Machine Learning System for a Practical Problem:

A case study of the hotel industry openings in new locations

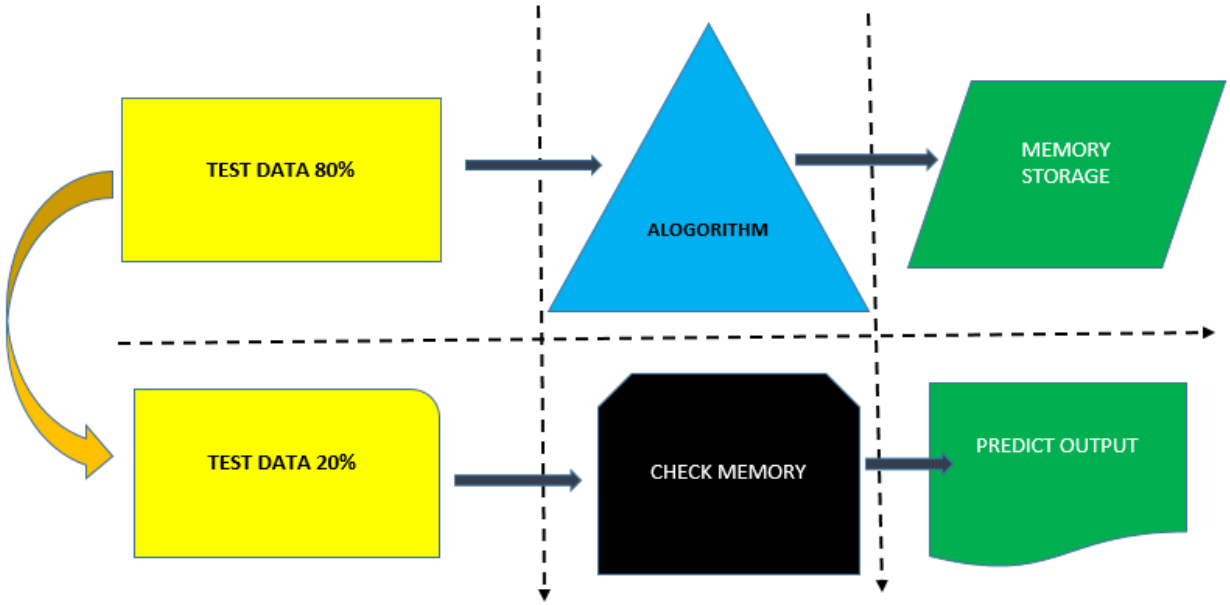
Introduction

Data is becoming the core central performance of every organisation and application systems. Developers and engineers are consistently coming up with algorithms and software products that are tailored to solving the existing organisation problems. One of the existing problems that corporations are fighting to solve today are data related. Most corporations are producing a lot of data, in fact so much that they can't consume within the allocated time and periods. So this is where data security comes in. Data security takes into considerations all the factors needed to ensure that the propose system, in both theory and practise is adaptable to the existing risking from both its internal surface and the external alienation.

However, as part of a major operative carried out by most banks, of the most of the factors that are concerned with drawing inferences from the datasets that are acquired by the organisations. Several departments like Human resource, supply chain and procurement, Information technology and marketing acquire a lot of data that the company can use in ensuring that the organisation is steered into the right direction. At this point, the role of data engineers and scientists come into play. Since management consistently requires regular updates and insights onto the existing departmental interactions within the organisation, data engineers have gone ahead to develop platforms and software applications that basically support this role and idea, they have done this by developing Business intelligent applications that robustly consume data, apply algorithms to it and then beautifully apply presentational graphs and linear visuals on the analysis to help the organisations make informed decisions. On top of the added functionalities provided by most analytical tools and software, the need for predictive analytics has also risen and companies want to know how much they should expect in future.

For instance, in case of a risk, how much is enough risk and to what extent should the risk go. The materiality of the risk entirely depends on how well the company has predicted the future reliance of the current state of affairs within the company given the current dataset and the future datasets. Predictive analytics relies in a core concept in the A/B rule of train and test. For 80% of the data supplied should be training purpose and the other 20% for the purposes of testing. During the training phase, the algorithm learns based on the dataset supplied, the

learning curve helps the algorithm to identify what would be the most expected ideal scenario of on input. Consider the situation below:



From the above illustration, it can be observed that the 80% data supplied into the algorithm is specifically split for training and then the resultant learns are store in the computer memory. Later, when another test is supplied into the memory, it checks from the same memory and then reads the exact associated rule and the prints the required output from the table.

Problem statement

The existing hotel company needs a new machine learning model and algorithm to identify these best way forward to help it know if the next hotel branch opened in another part of town is going to profitable or not. For this particular exercise, in order to achieve this, the identification of the dataset existing and the machine learning algorithms amiable is going to help quantify, match and predict the next chain of hotel store profitability based on the output of the dataset.

Objectives and aims

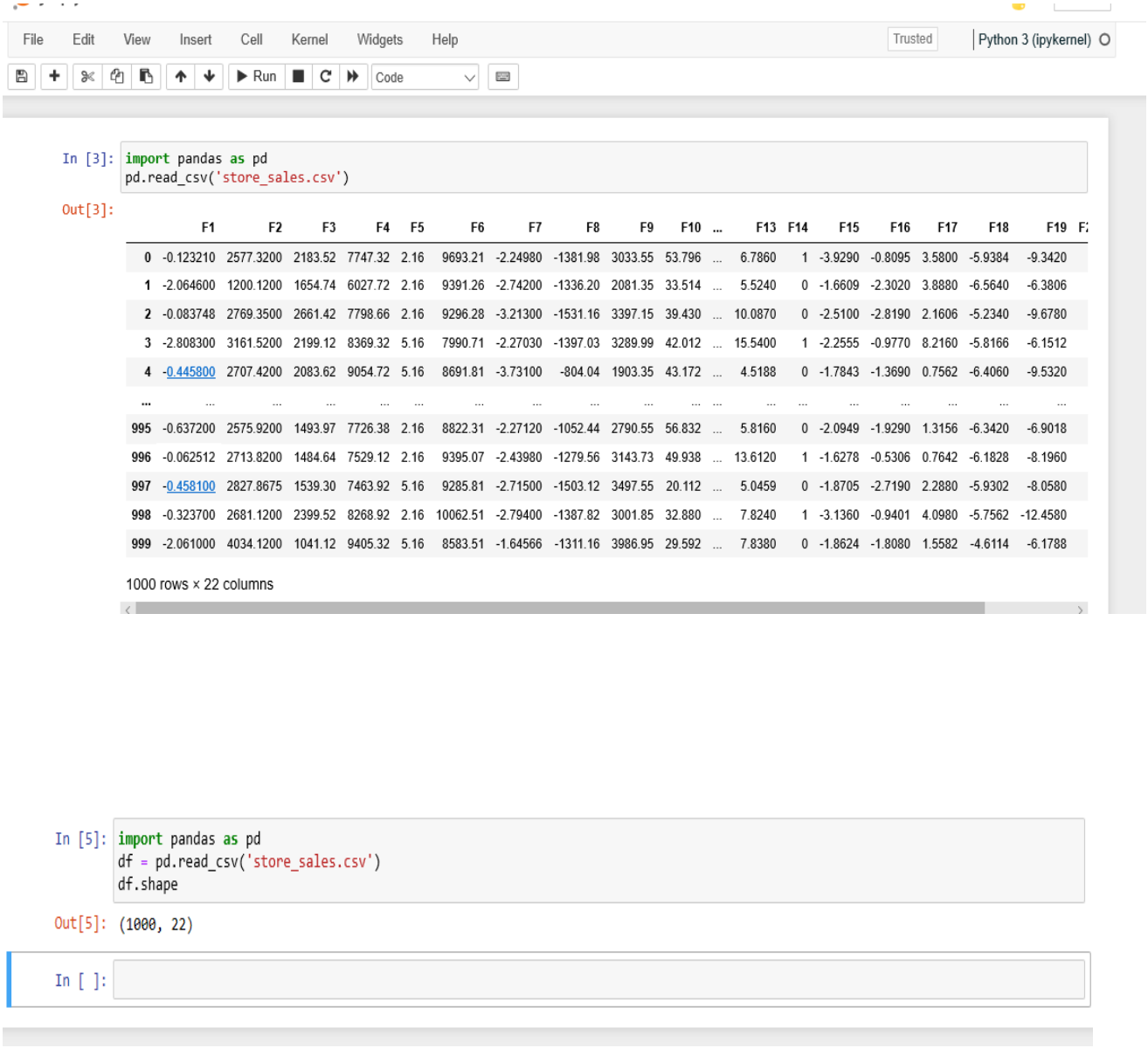
- To learn to identify machine learning techniques appropriate for a particular practical problem
- To undertake a comparative evaluation of several machine learning procedures when applied to the specific problem
- To predict the possibility of a new hotel branch being profitable in a new town

Research methodology and design

The analysis of this dataset is based on the python program and the Python program Jupyter Notebook. All code run on Python Jupyter Notebook web and interpreted. The stores dataset contains a list of 1000 records of store locations and their potential profits in the different locations that they are operating.

Analysis:

A preview of the dataset on Jupyter looks as follows:



```
In [3]: import pandas as pd
        pd.read_csv('store_sales.csv')
```

Out[3]:

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	...	F13	F14	F15	F16	F17	F18	F19	F2
0	-0.123210	2577.3200	2183.52	7747.32	2.16	9693.21	-2.24980	-1381.98	3033.55	53.796	...	6.7860	1	-3.9290	-0.8095	3.5800	-5.9384	-9.3420	
1	-2.064600	1200.1200	1654.74	6027.72	2.16	9391.26	-2.74200	-1336.20	2081.35	33.514	...	5.5240	0	-1.6609	-2.3020	3.8880	-6.5640	-6.3806	
2	-0.083748	2769.3500	2661.42	7798.66	2.16	9296.28	-3.21300	-1531.16	3397.15	39.430	...	10.0870	0	-2.5100	-2.8190	2.1606	-5.2340	-9.6780	
3	-2.808300	3161.5200	2199.12	8369.32	5.16	7990.71	-2.27030	-1397.03	3289.99	42.012	...	15.5400	1	-2.2555	-0.9770	8.2160	-5.8166	-6.1512	
4	-0.445800	2707.4200	2083.62	9054.72	5.16	8691.81	-3.73100	-804.04	1903.35	43.172	...	4.5188	0	-1.7843	-1.3690	0.7562	-6.4060	-9.5320	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	-0.637200	2575.9200	1493.97	7726.38	2.16	8822.31	-2.27120	-1052.44	2790.55	56.832	...	5.8160	0	-2.0949	-1.9290	1.3156	-6.3420	-6.9018	
996	-0.062512	2713.8200	1484.64	7529.12	2.16	9395.07	-2.43980	-1279.56	3143.73	49.938	...	13.6120	1	-1.6278	-0.5306	0.7642	-6.1828	-8.1960	
997	-0.458100	2827.8675	1539.30	7463.92	5.16	9285.81	-2.71500	-1503.12	3497.55	20.112	...	5.0459	0	-1.8705	-2.7190	2.2880	-5.9302	-8.0580	
998	-0.323700	2681.1200	2399.52	8268.92	2.16	10062.51	-2.79400	-1387.82	3001.85	32.880	...	7.8240	1	-3.1360	-0.9401	4.0980	-5.7562	-12.4580	
999	-2.061000	4034.1200	1041.12	9405.32	5.16	8583.51	-1.64566	-1311.16	3986.95	29.592	...	7.8380	0	-1.8624	-1.8080	1.5582	-4.6114	-6.1788	

1000 rows × 22 columns

```
In [5]: import pandas as pd
        df = pd.read_csv('store_sales.csv')
        df.shape
```

Out[5]: (1000, 22)

```
In [ ]:
```

Using the def (), it was possible to show the records in the dataset and how many columns are part of the dataset. In this case, it was a total of 1000 records and 22 columns of data.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	-1.512791	2520.923949	2120.448916	8459.648266	3.747000	8571.085469	-2.344597	-1249.307215	3205.505415	36.836583
std	2.015813	747.470842	1587.054314	2074.355688	1.498224	2108.672583	0.614178	493.077434	511.582397	9.723283
min	-17.514000	-2459.880000	-7321.080000	-655.080000	2.160000	-5873.790000	-5.111000	-4531.040000	-718.650000	5.420000
25%	-1.978575	2440.070000	1505.370000	7659.360000	2.160000	8291.085000	-2.724500	-1423.122500	3071.450000	30.700500
50%	-0.760800	2709.570000	1762.455000	8046.520000	5.160000	8986.140000	-2.210750	-1356.980000	3216.461500	36.759900
75%	-0.191055	2836.273250	2227.920000	8686.670000	5.160000	9414.442500	-1.857700	-1222.365000	3341.050000	42.986500
max	-0.060003	6563.120000	18814.920000	24004.920000	5.160000	19761.210000	-1.570592	1930.960000	6685.350000	75.180000

A summary of the column looked like below

```
In [7]: df.values
```

```
Out[7]: array([[-0.12321, 2577.32, 2183.52, ..., 1, nan, True],
               [-2.0646, 1200.12, 1654.74, ..., 1, nan, False],
               [-0.083748, 2769.35, 2661.42, ..., 0, nan, False],
               ...,
               [-0.4581, 2827.8675, 1539.3, ..., 1, 33.81, False],
               [-0.3237, 2681.12, 2399.52, ..., 0, 35.79, True],
               [-2.061, 4034.12, 1041.12, ..., 0, 37.8, True]], dtype=object)
```

For final machine model learning, the model was split by taking samples of this dataset that included the population of the area, the sales per shop and the status of the shop i.e. TRUE for profitable and FALSE for not profitable. The resultant dataset was put on test

In [17]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
store_data = pd.read_csv('store_v1.csv')
X = store_data.drop(columns=['profitable'])
y = store_data['profitable']

model = DecisionTreeClassifier()
model.fit(X, y)
predictions = model.predict( [  [2577.32,2183.52], [1680.12,2708.52] ])
predictions
```

```
C:\Users\2395648\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-packages\sklearn\base.py:445: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

Out[17]:
```
array([ True, False])
```

In [ ]:

In [ ]:

The predictions from the above show that with a sample input of 2577 population and a sale of 2183 daily revenue, the shop in this place would be making profits whereas the reverse is true when the population is 1680 and the sale is 2708 . Statistically, this two variables re not directly related to one another and are subjects to other external forces.