Warehouse Storage Optimization - Report 6

Dhruvil Dave
School Of Engineering And Applied Sciences
Ahmedabad University
Enrollment No.: AU1841003
email: dhruvil.d@ahduni.edu.in

Dhatri Kapuriya
School Of Engineering And Applied Sciences
Ahmedabad University
Enrollment No.: AU1841129
email: dhatri.k@ahduni.edu.in

Harvish Jariwala
School Of Engineering And Applied Sciences
Ahmedabad University
Enrollment No.: AU1841050
email: harvish.j@ahduni.edu.in

Nisarg Thoriya
School Of Engineering And Applied Sciences
Ahmedabad University
Enrollment No.: AU1841142
email: nisarg.t@ahduni.edu.in

***Abstract*—This is the sixth progress report of our group *Gopher - Group 5* for Machine Learning (CSE523) course project.**

***Index Terms*—Time series forecasting, classification, data preprocessing, clustering, graphical models**

I. Introduction

For our project, we decided to use the Amazon Bin Images Dataset.

This is originally a Computer Vision dataset. The Amazon Bin Image Dataset contains over 530,000 images and metadata from bins of a pod in an operating Amazon Fulfillment Center.

II. Brief insight of previous work

In the previous week, we converted the dataset from dictionary form to tabulated form. After tabulating, we expanded the dataset to 1 million rows and 20 columns by using CTGAN (Conditional Generative Adverserial Network) as we wanted data that represented other warehouses too. Then we finally moved on to fitting a Decision Tree classifier and we were able to obtain an accuracy of 89.43%

III. Task Performed And Outcomes

We noticed a mistake in the code this week in which when the object was not placed into the bin, instead of the dataset depicting bin number to be -1, the bin number remained the last successfully placed bin. After correcting that, our dataset changed to this:

	0	2	3	100	101	102	103	104	105	106	...	110	111	112	113	114	115
0	0.208461	True	0	0.791539	1.000000	1.000000	1.00000	1.00000	1.000000	1.0000	...	1.000000	1.000000	1.00000	1.000000	1.000000	1.000000
1	0.481681	True	0	0.309858	1.000000	1.000000	1.00000	1.00000	1.000000	1.0000	...	1.000000	1.000000	1.00000	1.000000	1.000000	1.000000
2	0.420538	True	1	0.309858	0.579462	1.000000	1.00000	1.00000	1.000000	1.0000	...	1.000000	1.000000	1.00000	1.000000	1.000000	1.000000
3	0.859182	True	2	0.309858	0.579462	0.140818	1.00000	1.00000	1.000000	1.0000	...	1.000000	1.000000	1.00000	1.000000	1.000000	1.000000
4	0.171162	True	0	0.138697	0.579462	0.140818	1.00000	1.00000	1.000000	1.0000	...	1.000000	1.000000	1.00000	1.000000	1.000000	1.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
999995	0.315423	False	-1	0.003609	0.034373	0.002928	0.00008	0.00027	0.013412	0.0038	...	0.023717	0.016353	0.05163	0.003678	0.002267	0.001295
999996	0.230940	False	-1	0.003609	0.034373	0.002928	0.00008	0.00027	0.013412	0.0038	...	0.023717	0.016353	0.05163	0.003678	0.002267	0.001295
999997	0.933347	False	-1	0.003609	0.034373	0.002928	0.00008	0.00027	0.013412	0.0038	...	0.023717	0.016353	0.05163	0.003678	0.002267	0.001295
999998	0.971583	False	-1	0.003609	0.034373	0.002928	0.00008	0.00027	0.013412	0.0038	...	0.023717	0.016353	0.05163	0.003678	0.002267	0.001295
999999	0.988404	False	-1	0.003609	0.034373	0.002928	0.00008	0.00027	0.013412	0.0038	...	0.023717	0.016353	0.05163	0.003678	0.002267	0.001295

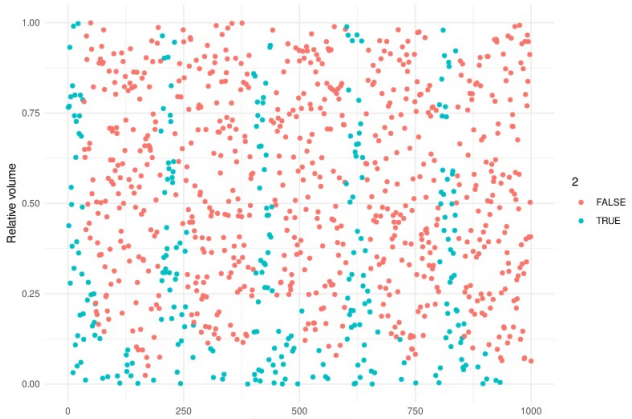
1000000 rows × 23 columns

After tweaking the dataset, we tried to apply classical Machine Learning algorithms including Decision Tree, Random Forest, k-Nearest Neighbours, Linear SVC. We obtained the following accuracies on using these models:

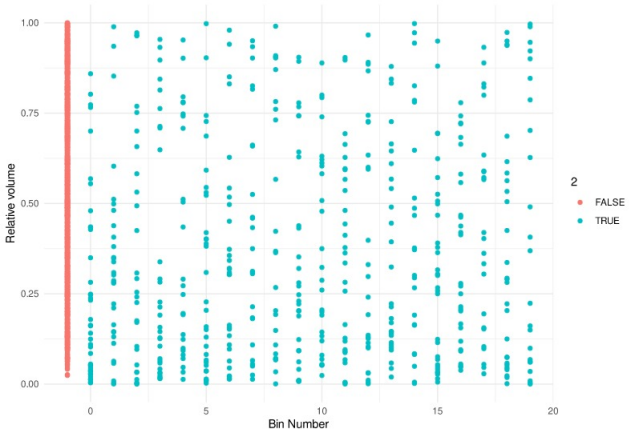
Algorithm	Accuracy
Decision Tree	82.46%
Linear Support Vector Classified (C=1)	82.69%
Linear Support Vector Classified (C=0.1)	82.82%
KNN (n_neighbours = 3)	83.41%
Random Forest (n_estimators = 20)	83.97%
Random Forest (n_estimators = 50)	84.05%
Random Forest (n_estimators = 100)	84.14%
KNN (n_neighbors = 5)	84.23%

As we can see, on solving the bug of the dataset, out accuracy on the decision tree reduced from 89.43% to 82.46%

Some of the plots that we obtained during feature engineering are as given below The below graph shows that there no particular pattern in placement of the products.




```
ggplot(df[20001:21000, ]) +
  theme_minimal() +
  xlab("Sequence Number") +
  ylab("Relative volume") +
  geom_point(aes(x = seq(1000), y = `0`, color =
    `2`))
```




```
ggplot(df[140001:142000, ]) +
  theme_minimal() +
  xlab("Bin Number") +
  ylab("Relative volume") +
  geom_point(aes(x = `3`, y = `0`, color = `2`))
```

IV. Tasks for Upcoming Week

The main tasks to be performed in the upcoming week are:

1) Exploring time series modelling
2) Tuning the hyper-parameters of the model