Warehouse Storage Optimization

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*—In this project, we decided to experiment with a real world dataset, and to explore how machine learning algorithms can be used to find the patterns in data. We chose Amazon Bin Images Dataset for this project. After performing the required tasks on a dataset we have tried implementation of few models on the dataset.**

***Index Terms*—Classification, Data pre-processing, Synthetic data, Time series forecasting, Decision tree, K-Nearest Neighbors Classifier, Random Forest Classifier**

I. Introduction

The main problem statement of our project is to solve the problem of Warehouse Storage Optimization using Machine Learning [1]. Often it can be solved with the help of classical optimization theory and sorting problems but using machine learning explores an entirely new outlook to the problem.

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. We used the meta data of this data set and then

II. Literature Survey

Our goal in this project was to collect amazon bin data and provide best possible optimising machine learning algorithms for its storage. Hence, we reviewed certain algorithms and techinques taking place in the current scenario.

A. Optimising Warehouse Operation with Machine Learning on GPU

In 2013, Nvidia attempted to tackle this problem by using a Picker Routing problem. This is a special case of the Travelling Salesman problem. In this algorithm, we try to find out the most optimised path the worker should take across the warehouse after which, he has the least amount of items left with him. They came up with an OCaPi (Optimal Cart Pick) algorithm to solve this. [2]

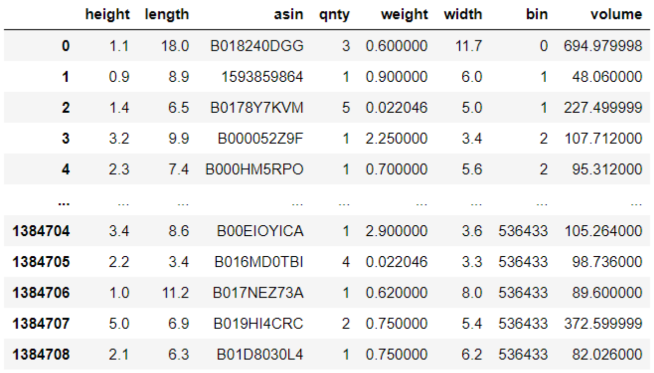
B. Predicting Warehouse Storage

Analytics Lab of MIT attempted to optimise this problem in 2017. They employed many different machine learning models like Linear Regression, Gradient Boosting and fully connected neural network. They obtained a 32%, 48%, 58% and 79% acceptable prediction values in Naive Bayesian, Gradient Boosted Trees, Text Enhanced Linear Regression and 3-layer NN. (Acceptable predictions are defined as prediction that are within 200% of true value and/or withinn 10 days of true value)

III. Implementation

For implementation of this storage optimisation problem we collected the metadata files of the Amazon bin images. These metadata files contained information about the image of the storage bin in which the products were stored. The most important task was to perform **Exploratory Data Analysis**. Since we were not going to use any of the deep learning techniques, we decided to drop the images entirely from our dataset and here we were left with alomst 530,000 JSON files to be cleaned and preprocessed. We wrote a simple python script to automate the process of parsing the json file and updating the database with the fields. After this was done, we had a database that contained JSON blocks corresponding to each image. Now that is clean enough to work with. The data consists of seven columns each explaining the product features. They are height, length, asin(Amazon Standard Identification Number), qnty, weight, width and bin. After cleaning and merging all the metadata files we shaped the data using Pandas [3] and Numpy [4]. Here, we assume that the bin size are equal and the volume column defines the occupied space in the respective bins.

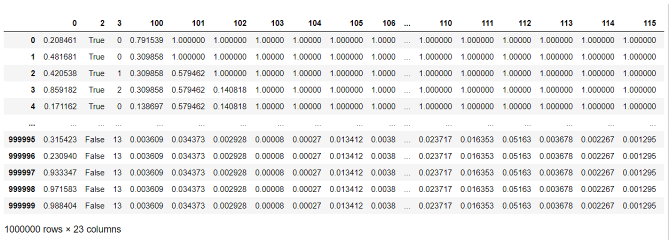
We had a dataset with a single pattern arrangement. Machine learning model can performed more accurately with varied data. Hence, we created a synthetic dataset with the help of CTGAN. This dataset consists of 23 columns. These columns consists of volume of each input product,information if it has been placed or not, bin number in which it has been placed



	height	length	asin	qnty	weight	width	bin	volume
0	1.1	18.0	B018240DGG	3	0.600000	11.7	0	694.979998
1	0.9	8.9	1593859864	1	0.900000	6.0	1	48.060000
2	1.4	6.5	B0178Y7KVM	5	0.022046	5.0	1	227.499999
3	3.2	9.9	B000052Z9F	1	2.250000	3.4	2	107.712000
4	2.3	7.4	B000HM5RPO	1	0.700000	5.6	2	95.312000
...	...	...	...	...	...	...	...	...
1384704	3.4	8.6	B00EIOYICA	1	2.900000	3.6	536433	105.264000
1384705	2.2	3.4	B016MD0TBI	4	0.022046	3.3	536433	98.736000
1384706	1.0	11.2	B017NEZ73A	1	0.620000	8.0	536433	89.600000
1384707	5.0	6.9	B019HI4CRC	2	0.750000	5.4	536433	372.599999
1384708	2.1	6.3	B01D8030L4	1	0.750000	6.2	536433	82.026000

Fig. 1. The dataset we obtained from amazon with added volume

and the vacant space left in each of the 20 bins. We have such 1 millions products i.e. a dataset with dimensions 1000000 x 23.



	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	13	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	13	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	13	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	13	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	13	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

We split this dataset into 95% train and 5% test which would give us 9,50,000 samples for train and 50,000 samples for test.

IV. Results

We then trained a few algorithms viz. Decision Tree, Random Forest, K-Nearest neighbours on this synthetic data set using Scikit learn [5]. The scores of these algorithms can be seen in the table below:

Algorithm	**Accuracy**
KNN (n_neighbours = 3)	55.80%
KNN (n_neighbors = 5)	50.49%
Decision Tree	89.7%
Random Forest (n_estimators = 20)	90.89%
Random Forest (n_estimators = 50)	91.28%
Random Forest (n_estimators = 100)	91.39%

V. Conclusion

Up until now, we have spent a significant amount of time in making our dataset tidy so various types of models can be applied. We also started with initial modelling phase. There is still a lot to be done. Our main aim to be done further would be to apply various kinds of possible classification models including the time series ones and doing hyperparameter tuning as well.

References

[1] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," *arXiv preprint arXiv:2002.04803*, 2020.

[2] K. Roodbergen and R. De Koster, "Routing order pickers in a warehouse with a middle aisle," *European Journal of Operational Research*, vol. 133, no. 1, pp. 32–43, 2001.

[3] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[4] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.