PROJECT GROUP 2

Abilash Nair
Department of Data Science
Indiana University
Bloomington, IN 47403
abinair@iu.edu

Mridul Chavan
Department of Data Science
Indiana University
Bloomington, IN 47403
mrichav@iu.edu

Shanmukha Surapuraju
Department of Computer Science
Indiana University
Bloomington, IN 47403
shasurap@iu.edu

Akhil Nagulavancha
Department of Intelligent Systems
Indiana University
Bloomington, IN 47403
aknagu@iu.edu

December 2, 2019

ABSTRACT

The domain of this project is confined to the retail business and with this project we are trying to tackle three important question which pertain to the DSD retail domain. We did extensive data processing and exploratory data analysis to remove redundant features and create relevant ones. We would be using regression, multi-class classification and clustering algorithms to make the predictions to obtain answers for these questions.We were able to achieve good enough performance from the above mentioned models and were able to draw inferences which could help improve the performance even further.

Keywords Routes · Keras · Regression · Multi-Class Classification · Clustering

1 Introduction

Direct Store Delivery accounts for more than 20 million dollars' worth of retail product distribution every day. DSD also accounts for more than 52 percent in retail profits. Since its heavily used for delivering perishable products, it becomes extremely important to plan and maintain delivery schedules to get products on the shelves and to pick up unsold products. Also, it is essential to carefully plan while incorporating stores into existing warehouses or expanding into new territories and setting up new warehouses. With this project we try to address 3 important questions related to the above-mentioned process - predicting the time required for the completion of a route, the right place to put a warehouse and finally, assigning a store to the right warehouse.

To understand the first question, there needs to be some explanation regarding what a route is and how deliveries are done. A route is nothing but a delivery schedule. The delivery schedules in DSD are made by product manufacturers for the orders placed by retail stores. When a driver goes on a route, he or she must go from the warehouses to multiple stores and deliver or pickup products from these stores. So, a route is basically a list of store visits. A loader and assembler accompany the driver on this trip, to help them with, as the name would suggest, loading and assembling the products in the truck.

*.

2 THE DATASET –

We started off the project with the following datasets –

1) ROUTE DATA

ID	Actual Start	Actual End	DRIVER ID	LOADER ID	ASSEMBLER	Vehicle	Territory
9751300686	9/27/2019 11:36	9/27/2019 17:57	12329623	LA12331966	LA12331607	11033 (Truck)	W1010

The route data consisted the following columns –

a) ID – The ID of the route

b) Actual Start time – The start time logged in the application by the driver

c) Actual End time – The end time logged in the application by the driver

d) Driver ID – The employee ID of the driver

e) Assembler ID - The employee ID of the assembler

f) Loader ID – The employee ID of the loader

g) Vehicle ID – The ID of the vehicle

h) Territory ID – The ID of the warehouse from which the products are loaded

2) STORE DATA

ID	Address	City	State	Territory
729541000	1510 N BEACH ST	FORT WORTH	TX	W0163

The store data consisted of the following columns –

a) ID – The ID of the store

b) Address – The address of the store

c) City – The city in which the store is located

d) State – The state in which the store is located

e) Territory – The warehouse to which the store is assigned

f) Date - Date when the store was inducted into a warehouse

3) STORE VISIT DATA

Store ID	ID	State	Route ID	Description
912140102	C067258533	NJ	3959630069	(Delivered) 3/30/2019 19:02 221 SHOPRITE STORE 221 5.6

The store visit data consisted of the following columns -

a) Store ID – The ID of the store for which the visit was assigned

b) ID – ID of the store visit

c) Route ID – ID of the route, of which the visit will be a part of

d) Visit description – The name of the store and the visit data and time description

e) State – The state in which the store visit was active

2.1 FEATURE EXTRACTION

1) For calculating the distance covered by the driver and to predict the time that they'll take, we needed to know the location details like latitude and longitude. We wrote a program which would use the address data for each store and find the latitude and longitude for each of these stores.

2) After obtaining the location details, we added another column which represented the distance travelled by the driver while on the route.

3) Since the time taken for the entire delivery is also dependent on the skill level of the driver, loader and assembler, for each route, we calculated the number of trips that the driver, loader and assembler had made before that route.

4) Using the value calculated in terms of experience, we gave the drivers, loaders and assemblers a potential score (since we did not know how much experience they had in the beginning) and incremented it by 0.5 for every ride to create the experience scores.

5) We then used the start time of the route to create categories to know at what time of the day did the route start, since traffic conditions are different during different times of the day. Using excel, we had rounded off the start time to the nearest hour and then we wrote a script to distribute the data into 3 categories – 00:00 – 7:00. 8:00 – 20:00 and 21:00 – 24:00

6) We also wrote a script to calculate the time taken for each route, by subtracting the start time from the end time and rounding it off to the nearest hour.

7) For the stores data required for the second model, we first wanted to calculate the distance of each store from the nearby warehouses. Since we were looking at data for 3 states – Illinois, Indiana and Michigan, we had 10 warehouses in these states. So, we created one column each to show the distance of a store from each warehouse.

8) We also calculated the number of stores associated with each of these 10 warehouses within 200 km radius of every store.

9) Since, we already have information about the date the stores were inducted into the warehouses, we also created a column each for the number of stores that were present in each of these warehouses while the store was being inducted.

2.1.1 EXPLORATORY DATA ANALYSIS –

1) While going through the Route dataset, we found that there were routes which had start and end times that were only separated by 5 – 10 minutes, we had to remove these routes from the final data.

2) We also found routes that had a difference between their start and end times as large as 200 hours while only travelling a considerably shorter distance. These routes were removed as well.

3) A script was used to remove the two outliers stated above.

4) On inspecting the store data we found out certain stores which were not assigned to any particular territory, these stores were removed from the dataset.

5) Also, there were stores which had missing or invalid address lines, these stores were also removed from the dataset.

2.2 QUESTION 1 :

Problem Statement –

We want to predict the time required, in terms of number of hours, for completing a route.

Solution –

Using the above-mentioned feature generation steps, we created the dataset that is used to make the predictions by fitting the data to various models.

The final dataset has the following features –

a) Distance – The distance covered on the route

b) Driver Score – Experience score calculated for the driver

c) Loader Score – Experience score calculate for the loader

d) Assembler Score – Experience score calculate for the assembler

e) 0 to 7 – If the route started between 00:00 and 7:00

f) 8 to 20 – If the route started between 8:00 and 20:00

g) 21 to 23 – If the route started between 21:00 and 23:00

h) Time - The time taken to finish the route

2.3 Models Used and scores for test and train –

1) ScikitLearn Linear Regression - 0.9346321623816782, 0.9338965388667487

2) ScikitLearn Lasso Regression - 0.9346163329161463, 0.9339362531878

3) ScikitLearn Ridge Regression - 0.934632164132487, 0.9338954512180081

4) ScikitLearn Random Forest - 0.9880434131681542, 0.9206486475145964

2.4 DISCUSSION -

Most models which were run on the dataset gave considerably good scores. Although the following issues could be attributed as reasons for the erroneous predictions –

1) The start and end times are logged by the drivers using a mobile application and due to differing usage patterns, the drivers tend to log on and off at different stages of the delivery schedule. Some drivers tend to log off after completing the entire route and returning to the warehouse, while some tend to do it after finishing the last store on the schedule.

2) Drivers sometimes might tend to take different routes, which might be different than the ones that we considered while calculating the distance. Sometimes it is better to take a longer route through less traffic than a shorter one with clogged lanes.

3) Sometimes delivering at certain stores is different than others because of activities like taking proof of deliveries and taking cheques from cash customers. This amounts to more time than a usual stop and deliver activity at a store.

2.5 CONCLUSION -

We were able to create multiple models which could fairly predict the delivery times down to the nearest hour. The above-mentioned issues could be fixed with a standardized usage of the applications used to log in start and end times and by collecting odometer values from the vehicle or using a GPS navigation device to keep tabs on the distance. With these measures we will be able to get a better accuracy on our predictions while using the same techniques.

2.6 QUESTION 2 :

Problem Statement –

We want to identify which warehouse a store needs to be assigned to.

Solution –

Using the above-mentioned feature generation steps, we created the dataset that is used to make the predictions by fitting the data to various models.

The final dataset has the following features –

1) Lat – Latitude of the store

2) Lng – Longitude of the store

3) $WXXXX_D * 10$: Distance of store 10 nearby warehouses

4) $WXXXX_S * 10$: The number of stores under each of the 10 warehouses within 200kms

5) $WXXXX_R * 10$: Number of stores already under each branch while the store was being inducted

6) $WXXXX_Y * 10$: The target value, one column for each of the 10 warehouses.

We used a Keras Classifier from the keras library to make use of a neural network consisting of just one hidden layer with 22 neurons, which has a relu activation function and an output layer of 10 neurons with a softmax activation function. The loss function that we used is categorical cross entropy and the optimization method used is ADAM. Kfold cross validation was used to check the accuracy of the model.

2.7 Models Used and scores for test and train –

The metric used to measure the performance of the model is accuracy

Keras Classifier : 77.41 percent (S.D : 1.26 percent)

2.8 DISCUSSION -

The accuracy scores that we achieved with the Keras classifier was a respectable 77.41 percent, which considering the data that we started out with looks good. Initially we ran the model with all the extracted features and found that the accuracy was not as much as we expected and lingered around 30 – 40 percent, so we dropped one set of features which represented the number of stores that were already present in each branch while the store was being inducted and we found that the accuracy almost doubled. The erroneous predictions can be because of the following reasons –

1) There are more factors that come into play other than distance and number of stores in the neighborhood, when it comes to assigning a store to a warehouse. If the manufacturer has multiple products, and if a store only needs certain products then it might be assigned to a certain warehouse.

2) The frequency of the orders placed by these stores also come into play. Some stores place very few orders in a month and hence do not add much load to an already full warehouse.

2.9 CONCLUSION -

The model that we used for this multi class classification gave us a decent enough accuracy, one which can be increased with addition of some more features which represent the scenarios that we explained in the section above. We also could use confusion matrix to describe the performance of the model used for classification since we could have some

inherent imbalances in the dataset that we used.

2.10 Question 3 -

- In this question we identify the best possible location a warehouse can be setup using the location of the stores
- For this we are finding how stores can be clustered and determine the best location for setting up a warehouse.
- We have explored various algorithms like K Nearest Means, Artificial bee colony algorithm and Density based spatial clustering of applications with noise (DBSCAN) and decided to use DBSCAN as we can prioritize the setup of warehouse in the clusters with maximum density of stores.
- We have our store location data in the form of physical addresses, and we have written our program to convert the physical address into longitudes and latitudes.

Store location data in the form of physical address

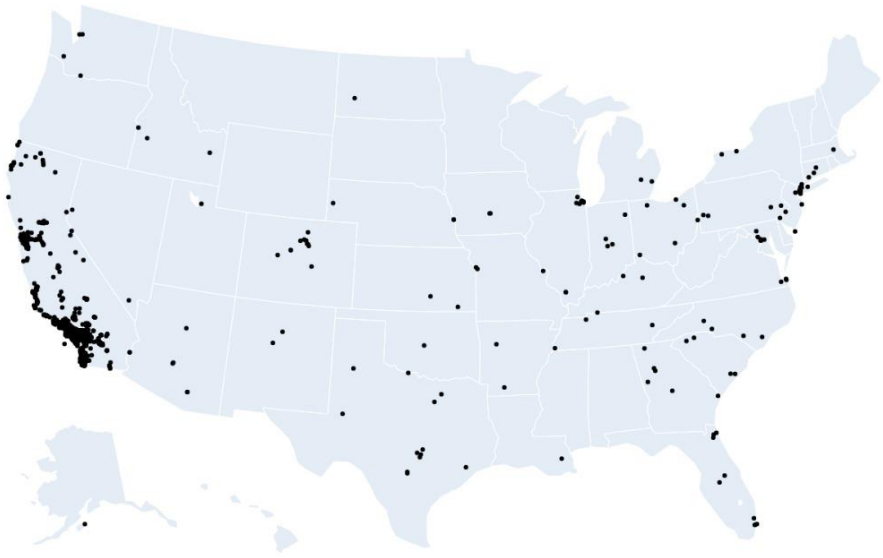
B	C	D
2950 MERCED ST	SAN LEANDRO	CA
2950 MERCED ST	SAN LEANDRO	CA
2950 MERCED ST	SAN LEANDRO	CA
2950 MERCED ST	SAN LEANDRO	CA
2950 MERCED ST	SAN LEANDRO	CA
2950 MERCED STREET	SAN LEANDRO	CA
30111 THOMAS	RANCHO SANTA MAR	CA
30111 THOMAS	RANCHO SANTA MAR	CA
322 LITTLEFIELD AVE	SOUTH SAN FRANCIS	CA
3575 REED AVE	WEST SACRAMENTO	CA
4075 LAKESIDE DR	RICHMOND	CA
4075 LAKESIDE DR	RICHMOND	CA
4075 LAKESIDE DR	RICHMOND	CA
4075 LAKESIDE DRIVE	RICHMOND	CA
4400 FLORIN PERKINS	BERKELEY	CA
4560 LOMA VISTA AVE	VERNON	CA
4560 LOMA VISTA AVE	VERNON	CA
4560 LOMA VISTA AVE	VERNON	CA
4560 LOMA VISTA AVE	VERNON	CA
4560 LOMA VISTA AVE	VERNON	CA
4578 E 49TH ST	VERNON	CA
490 ECCLES AVE	SOUTH SAN FRANCIS	CA
5 DORMAN AVE	SAN FRANCISCO	CA
5343 W IMPERIAL HWY STE 700	LOS ANGELES	CA
570 HARBOR SCENIC WAY	LONG BEACH	CA
570 HARBOR SCENIC WAY	LONG BEACH	CA

Store location data in the form of longitude and latitude coordinates

F	G	H	I	J
CA	25.75468	80.25476		
CA	25.77957	80.13477		
CA	26.08017	80.23901		
CA	27.84084	15.45958		
CA	28.02261	81.73335		
CA	28.30315	81.40828		
CA	28.67661	7.539013		
CA	29.37348	98.47651		
CA	29.43575	98.47887		
CA	29.68277	95.29513		
CA	29.93186	90.06656		
CA	30.14668	97.81484		
CA	30.16129	81.70189		
CA	30.27457	97.7616		
CA	30.30146	81.65712		
CA	30.34668	97.96619		
CA	30.36583	81.4938		
CA	30.51094	97.66783		
CA	32.03696	102.191		
CA	32.07191	81.09121		
CA	32.24597	110.9813		
CA	32.24598	110.9822		
CA	32.54669	117.0402		
CA	32.55397	116.9579		
CA	32.55397	116.9579		
CA	32.55464	117.0514		
CA	32.55954	116.9299		
CA	32.56802	117.1057		
CA	32.57013	117.0046		
CA	32.57594	117.0706		
CA	32.57808	117.0935		
CA	32.58164	117.0355		
CA	32.58312	117.0347		
CA	32.58491	117.0901		
CA	32.58692	117.0892		
CA	32.58985	116.9221		
CA	32.60248	117.0828		
CA	32.61405	117.0711		

We are considering longitude and latitude coordinates of the stores obtained, the features of DBSCAN algorithm so that we cluster the stores according to the location data.

The stores are visualized on Map as below.



DBSCAN clusters the data based on the density of the points and forms clusters from them considering the minimum number of points in the cluster and the maximum distance between the points in the cluster. We considered the Map for the region by considering the maximum and minimum longitude and latitude for proper plotting of clusters.

From the DBSCAN algorithm we obtained clusters of the stores according to their density in that area considering the minimum number points required to be in each cluster and the maximum distance between the points in each cluster as inputs. Our data is now labelled with the clusters based on density with DBSCAN algorithm. We have used scikit-learn's DBSCAN clustering package for implementation.

Once we have labeled our data we formed separate lists for longitude and latitude of the respective clusters and calculated centroids using them.

We consider centroids of that clusters to be the best place around which our warehouse can be built for the stores belonging to that cluster.

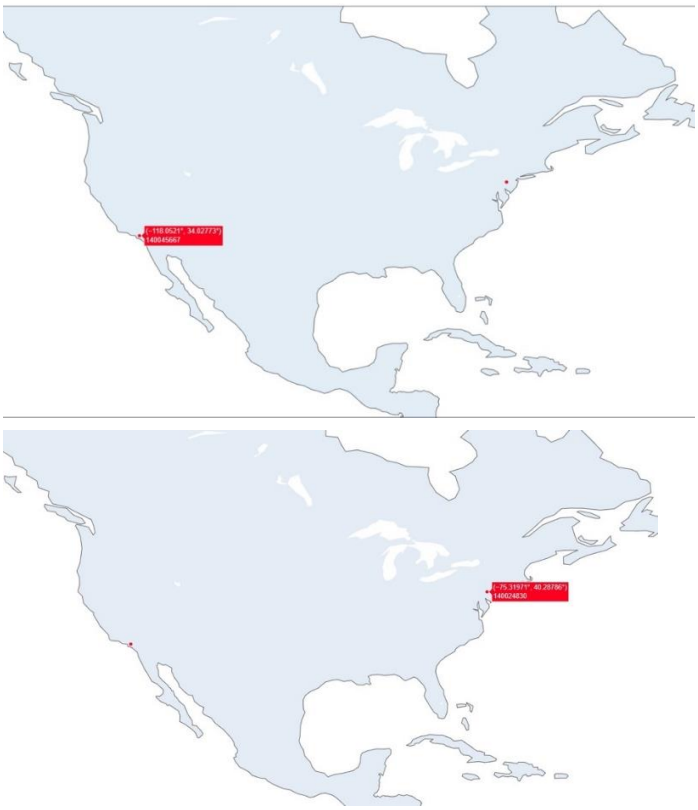
For the data we have considered, we have obtained two clusters and below mentioned points are the centroids for the two clusters formed. These coordinates seem to be the best location around which the next warehouse can be proposed.

2.11 Results -

Warehouse 1 - (-75.3197130451613, 40.2878640516129)

Warehouse 2 - (-118.05211866608549, 34.02773298386625)

The centroids of the obtained clusters are visualized as below



Basemap Matplotlib is the toolkit for plotting the clusters on the map. Basemap is the python tool for projection and visualization of geographical data on the Map. Since we are dealing geographical data with longitude and we have opted for Basemap matplotlib has it gives the best way to visualize the data. Required libraries required for Basemap are installed. We have found better alternative for visualizing the location data later and have used the plotly python library for visualizing the location data.