Basket – Shopping Elf App

Aditya Parmar, Dhrumil Shah, Neil Thaker, Rashmi Sharma

#*Software Engineering, San Jose State University*
1 Washington Sq., San Jose, CA 95192
rashmi.sharma@sjsu.edu
dhrumilnimish.shah@sjsu.edu
adityarajendrasinh.parmar@sjsu.edu
neil.thaker@sjsu.edu

ract— **This paper describes solution implemented for common le who frequently buy groceries. User can upload bill receipts on ile Application, which captures all data, stores and processes it to ide inventory notifications. Automated shopping list is generated machine learning algorithms on past data. Application enables to e-mail shopping list.**

ords— **unsupervised K-Means product clustering, multi-variate r regression, machine learning, optical character recognition, pt processing, open-source, San Jose state University**

I. INTRODUCTION

This document is a final report from our CMPE 272, Spring 7 Enterprise Software Platform as graduate students. Our tor, Prof Rakesh Ranjan directed our effort in this project for iding customer-centric solution and eventually, we came up this android application which can help customers to improve store shopping experience.

Architecture:
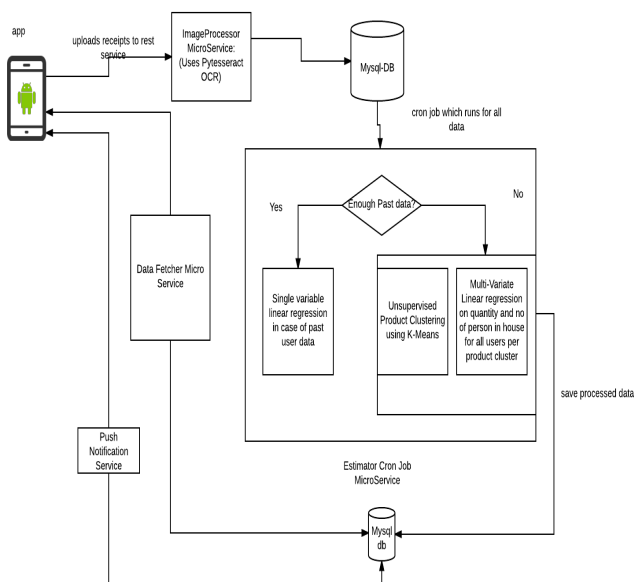
1) *Android Application:*

We have created Android application intuitive user interface which aids user in uploading receipts, viewing shopping list, inventory notifications.
We have used Retro-fit API for REST calls from Android application.

2) *Image Processing Micro-service:*

We have used open source py-tesseract Google OCR API to get content from uploaded image by the user. It is programmed in Python and coding from our side is also done in Python to get contents from API and store the data in MySQL database to further process and apply machine learning.
REST Services have been implemented and called by the android application for the continuous flow and give the output to the database. These services are deployed on AWS EC2 instance as a cloud solution and auto scaling.

3) *Machine Learning:*

We have built cron job which runs daily on all user receipts data and performs below process

i)   All receipt data per product for a user, is fed into linear regression algorithm, which takes quantity as feature and predicts estimated days for that product. Linear regression with one variable is only used in case optimal history data is available for that product.

$y = f(x)$ where x = quantity and y is estimated days.
$y = mx + c$

ii)   We have used unsupervised K-Means clustering for auto-clustering all products present in the system. Clustering happens using product name as feature and categorizes all the products.
Each product name is translated, stemmed, analysed using Natural Language processing Toolkit (nltk), which returns tokens.

Tokens from nltk processing is fed into K-Means clustering algorithm and clusters are formed.

iii) When optimal history data for product per user is not available, product's cluster is found from unsupervised clustering output and data of all users for all products of that cluster is fed to multi-variate linear regression to find estimated days per product category.

$Y = f(x_1, x_2)$
X_1 is quantity, X_2 is no of family members for that user.

This regression takes quantity and no of family members as features.
This calculates an average no of days this product is consumed considering all users.

Push Notification Service:

To remind users about the products which are low on inventory, we have integrated push notification service. For this, we have used "Firebase Cloud Messaging" Platform which gives the notification to customers. In the notification, we are listing all the products which are expected to get finish by the next day.

Starter & Cron scripts:

Application in Action:

1) *User signup*
2) *User logins with credentials*
3) *User uploads bill receipts, App provides cropping rotation etc. image editing features for better bill clarity.*
4) *User can view shopping list, which contains all the products which are about to end in next 6 days.*
5) *User can mail shopping list to registered email*

6) *User will receive daily notifications only for the products which are low on inventory.*

II. Conclusions

At first, out research experience on dividing problem space, designing application architecture we found many problems like How should we regress data? Which storage database would be good for storing inventory and processed data. While working on machine learning techniques we found many ways to find estimated days, but found linear regression fitting best the use case. Towards the end, we researched upon convolutional neural networks which can be used to further improve machine learning. While researching for Image processing OCR, we worked on Open-CV for image cropping and correction and then processed image using py-tesseract.
This project was quite a learning experience and has led us into exploring unsupervised classification and regression more.

References

[1] PyTessearct Library developed by Google. (https://github.com/madmaze/pytesseract)
[2] Python SciKit library (http://scikit-learn.org/stable/)
[3] Udacity_Machine_Learning_NanoDegree (https://www.udacity.com/course/machine-learning-engineer-nanodegree--nd009)
[4] Retrofit REST android api calling library (http://square.github.io/retrofit/)
[5] Firebase Cloud Messaging: Used for Push Notifications (https://firebase.google.com/docs/cloud-messaging)
[6] Mail sending (API: http://www.pythonforbeginners.com/code-snippets-source-code/using-python-to-send-email)