Online Retail Sales Intelligence Using Machine Learning

Suroor Hussain
Noufal Ibrahim
Asif ETV

Hamon Technologies LLP

http://hamon.in

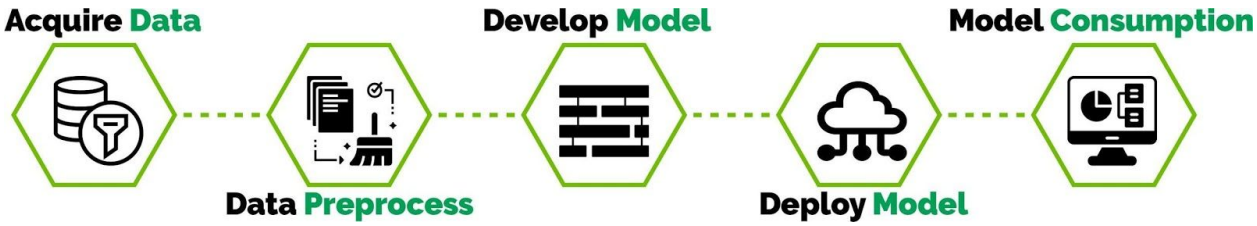
Online Retail Sales Intelligence Using Machine Learning

Introduction

The advent of internet into common man's life has brought many things online. From chats, calls and messages to food delivery, cabs and rentals, almost anything can be bought at the click of a button. The retail business is no exception. With many big and small players in the market and the ever growing opportunities available for anyone to step in at any time, the competition is also vast. Constant effort is needed to stand out and perform well consistently.

Our client is a company that uses sophisticated machine learning techniques to provide sellers with useful insights into their portfolio which will help them manage sales, plan, and perform better consistently. Hamon helped the client to predict future sales insights of using product details and past sales data if any. Insights such as how much a product needs to be priced so as to gain marginable profit. Multiple factors affect the sale of a product and in this paper we discuss the techniques and technologies used.

Hamon's Framework for Machine Learning



Hamon Technologies has developed a standard framework for solving complex AI problems that can be replicated across most of the machine learning projects we undertake. Clients come to us with various needs and based on their situation we may choose to use some or all of the steps in the framework.

In this project, We collected data by utilizing APIs provided by various sources such as online marketplaces. We developed custom data collectors and processed data to filter out broken and incomplete items. Filtered data was then sampled and used for training the model. The model was then deployed to an AWS SageMaker endpoint which was wrapped by a custom REST API for preprocessing of input features and easier consumption by the end application.

There were multiple predictors for various kinds of market intelligence for online retailing. This paper concentrates only on one component.

The prediction system

Data processing techniques were used to analyse years of online sales data and an ML model that accurately predicts sales estimate for a product based on various attributes was developed.

The technologies used in the design of the solution are:

- Scrapy, requests and BeautifulSoup
- Preprocessing techniques in sklearn and data manipulation using pandas
- Machine learning using sklearn
- Flask web framework
- Nginx HTTP server
- SQLAlchemy + PostgreSQL for storage
- Amazon S3 for raw data and archival

Acquire Data

To create a reliable and accurate prediction model, large amount of online sales data across many years was required. In addition to this we also used econometric data, climate data, and product attributes. The raw data was not completely structured. Some of it needed cleanup and augmentation and this was done before we could build the models. A pipeline to obtain, cleanup and augment the data was created which would

provide a feed of updated training data. The data obtained thus is stored into a database.

We needed to collect various attributes for large number of products across multiple online marketplaces such as Amazon, eBay and others. This was done using standardised APIs provided by these services which we consumed using Python libraries like requests. etc. For some data, we relied on third party plugins such as Unicorn smasher. Product data included color, pricing, monthly sales estimations etc.

This raw data was unstructured and varied according to source so we decided to use S3 as a persistence layer. We used the boto library to automate this last stage of the pipeline where we stored the data on S3.

The architecture was specifically designed to allow human oversight as well as easy accessibility from automation engines.

Data Preprocessing

The product and sales data acquired from running the Python scripts mentioned above and third party services were in different formats and structured differently as they were obtained from different sources. These needed to be cleaned up and formatted into a common format which would be later used for training the machine learning models.

The data cleaning and preprocessing also included deleting or augmenting products with no or negligible details as they could not be used for training the models. The biggest challenge was normalizing the data so as to it to be uniform. For example, products such as ‘Camera’s might fall under different categories, such as ‘Electronics’, ‘Cameras and Photos’ etc, on different online marketplaces.

To tackle this issue, Hamon decided on a number categories to work with and assigned them fixed names, such as ‘Clothing’ for textile related products, ‘Photography’ for etc. The preprocessor we designed, maps products from the e-commerce websites according to these category names. For example, products which fall in ‘Clothing and accessories’, ‘Clothing’, ‘Fashion’ and all similar were mapped into the same category name we decided upon ‘Clothing’.

Develop Model

The whole Machine Learning Model development was implemented using Amazon Sagemaker. SageMaker is a fully-managed service that covers the entire machine learning workflow to label and prepare the data, choose an algorithm, train the algorithm, tune and optimize it for deployment, make predictions, and take action. The whole project was deployed on AWS and t was only normal for Hamon to use Sagemaker which made the project easier for development and consumption.

One of the highlights of the Sagemaker is it allows its user to bring their own code through its 'Bring Your Own Code' (BYOC) service. The Sagemaker, over the already supported wide variety of Machine Learning algorithms, also supports user uploaded training and inference algorithms. These algorithms are packaged as Docker Images which gives the user the flexibility they need to experiment with different algorithms. Sagemakers flexibility enabled us to implement our own training algorithms and inference code.

We evaluated several algorithms and decided to use XGboost for regular predictions. XGboost stands for Extreme Gradient Boosting and is a gradient boosted decision tree algorithm. Initially training the XGboost with the whole dataset returned very low accuracy. Upon analysis, we found that the the marketplaces cater to different market and demographic segments which makes a single estimator unreliable. We built custom estimators for each marketplace (in one case even using a different algorithm) to get a higher overall accuracy. This approach gave us an aggregate accuracy of around 77%.

After an initial round of testing, we found that there are two classes of products. Ones which are recently introduced that don't have enough sufficient historical data to make accurate predictions and others which have

been in the market for some time. We developed a “light model” which was designed to work with a smaller feature set. The accuracy was lower (64%) but much higher than if we directly predicted with the rich model. After the product is in the market for a while, we’d gather historical data and switch to using the rich model.

Deploy Model

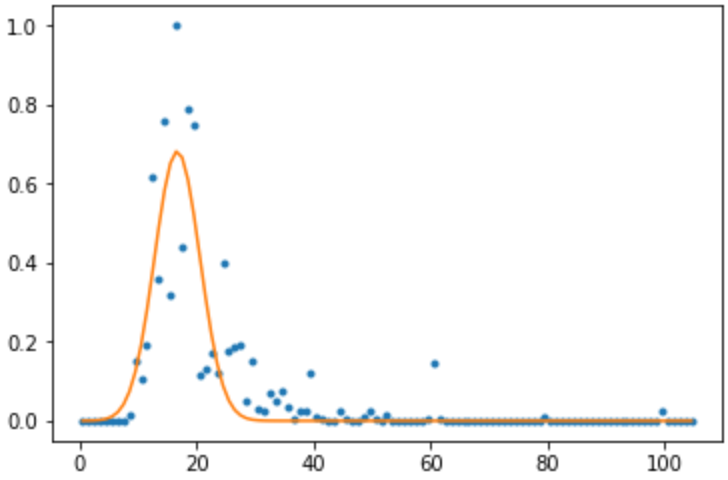
The next step is to deploy the trained Models for consumption. This was done using Sagemaker. As mentioned earlier, Sagemaker is an AWS service that makes creation, deployment, and management of machine learning models a first class activity on the AWS ecosystem. Before Docker, developers relied on heavy methods such as Virtual Machines. Docker containers on the other hand are lightweight as they run under a single operating-system kernel. It is used to create, deploy and manage virtualized application containers on a common operating system, with an ecosystem of allied tools. The container packages application services or function with all the supporting libraries, dependencies, configurations and other necessary parts required to run the application.

Hamon dockerized own training and inference algorithms and with the help of the Sagemaker developed models which can predict future sales of a product when fed with past sales and product information. These models were automatically pickled and saved to S3 by Sagemaker. Once deployed the Sagemaker provided us with endpoints that can be queried for predictions.

Retraining was done by reusing the Docker Images developed earlier and a new/modified dataset upon which new models and endpoints are created. This approach by Hamon not only made retraining easier than typical methods, but also allowed easy testing of new training algorithms.

Further services

The sales predictor was used to estimate several other entities such as optimal pricing. The predictor would tell us at what price, the sales would be maximum and we could offer that as a suggestion. This is briefly described below.



Fitting the Price vs Sales plot with a Gaussian Curve

The models give us a prediction of the number of units that will be sold in a month. Using these models, Hamon developed a service which will predict the price of the product for maximum profit. This was achieved by predicting the number of units sold within a price range of a product and calculating the profit from the predicted numbers and actual pricing.

Conclusion

Hamon designed, developed and deployed an Sales Insight prediction application which uses core python and Machine Learning techniques. The models have an aggregate accuracy of 70%. The application has been successful and smoothly deployed and is being consumed by the client.