Retail Sales Prediction and Application using Amazon Web Services

Jayesh Prasad Anandan (janandan@iu.edu)*, Afeefa Bano (abano@iu.edu)*, Navya Sree Santhapeta (nsantha@iu.edu)*

Abstract
To develop a machine-learning based application that predicts the retail sales of an enterprise with the help of XGBoost Algorithm implemented using AWS Sagemaker. Later, deploy the endpoint of this model as an API (Application Programming Interface) with the help of AWS Lambda and AWS API Gateway. Finally, to create a user-friendly webpage to allow users to input data and generate sales predictions.

Keywords
Amazon Web Services, Machine Learning, XGBoost, Cloud Computing

* *Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

Contents

1. Introduction

Techniques for making future predictions based on the present and past data have always been an area with direct application to various real-life problems. Sample historical sales data and other relevant information such as departments, holidays, stores, markdowns are gathered from the retail sales management system to forecast future sales for easy scalability and maintenance. Accurate sales forecasting can help businesses optimize operations, reduce costs, and improve profitability. The main objective of this project is to develop a machine learning (ML) predictive model that can generate accurate retail sales forecasts using AWS SageMaker.

2. Related work

1. *Dr. Bandaru Srinivasa Rao, Dr. Kamepalli Sujatha, Dr. Nannpaneni Chandra Sekhara Rao, Mr. T. Nagendra Kumar* in their paper *(Retail Sales Forecasting using Machine Learning Algorithms)* [1] proposed a retail sales forecasting model using machine learning algorithms, including XGBoost. The model was trained on a dataset of historical sales data and used several input features, including product attributes, promotions, and seasonal trends, to generate accurate sales forecasts.

2. *Devendra Swami, Alay Dilipbhai Shah, Subhrajeet K B Ray* in their paper *(Predicting Future Sales of Retail Products using Machine Learning)* [2] presents a study on using machine learning algorithms for predicting future sales of retail products. The authors investigate the application of several machine learning algorithms, including XGBoost, Random Forest, and Deep Neural Networks, for this task. They also explore various feature engineering techniques, such as the use of lag features and external data sources, to improve the predictive performance of the models.

3. *Amazon Web Services* in their blog post *(Call an Amazon SageMaker model endpoint using Amazon API Gateway and AWS Lambda)* [3] gives us an overview on how to deploy a machine learning model from AWS SageMaker as an API.

All these related works highlight the importance of accurate sales forecasting in retail and the potential benefits that can be gained from using machine learning techniques. The proposed project, Retail Sales Prediction Using AWS SageMaker XGBoost (Regression), aims to build upon these existing works by using AWS SageMaker's managed training and hosting platform to create a scalable and accurate predictive model for retail sales forecasting.

3. Method

3.1 Dataset

The dataset we used in our project is sourced from Kaggle, named *Retail Data Analytics* [4]. The dataset contains weekly sales from 99 departments belonging to 45 different stores with information on the holidays and promotional markdowns offered by various stores and several departments throughout the year.

This data consists of three sheets:

1. Stores: Anonymized information about 45 stores such as type and size of the store.
2. Features: Contains Additional data related to the store, department, and regional activity for the given dates such as
   (a) Store - The store number
   (b) Date - The week
   (c) Temperature - The average temperature
   (d) Fuel_Price - The Cost of fuel in the region
   (e) MarkDown 1-5 - Anonymized data related to promotional markdowns.
   (f) CPI - the consumer price index
   (g) Unemployment - the unemployment rate
   (h) IsHoliday - Whether the week is a special holiday week
3. Sales: Historical sales data, which covers from 2010-02-05 to 2012-11-01 contains the following:
   (a) Store - The store number
   (b) Dept - The department number
   (c) Date - The week
   (d) Weekly_Sales - Sales for the given department in the given store
   (e) IsHoliday - Whether the week is a special holiday week

3.2 Design and Framework

The Framework of our project consists of a Web Application with access to a Machine Learning Model endpoint which is deployed as API(Application Programming Interface). By making use of a user-friendly webpage, we allow users to input data and generate sales predictions.

The whole project utilizes the services provided by Amazon Web Services (AWS) Cloud. An AWS Account was created in the region US East (N. Virginia) us-east-1. Later Added team members as Identity and Access Management (IAM) users to the same account with required permissions for the services used in the project. Added Multi-Factor Authentication and additional security measures to restrict access to our account. The Free-Tier program in AWS provides access to various services for free with some limitations on usage.

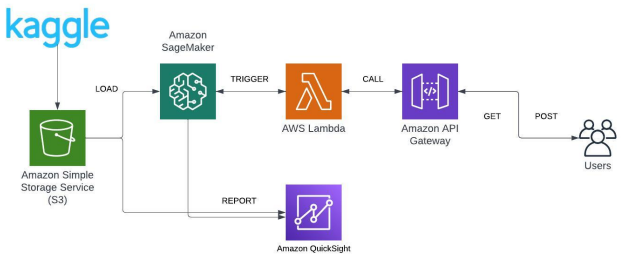

Figure 1. Project Workflow

1. Amazon S3
   (a) The Simple Storage Service (S3) in AWS offers object storage service with industry-leading scalability, data availability, security, and performance. The Free-tier gives 5GB of standard storage for free.
   (b) The dataset from Kaggle was uploaded to an S3 bucket named retail-sales-ecc as the name has to be unique.
2. Amazon SageMaker
   (a) Amazon SageMaker is a cloud machine-learning platform that enables developers to create, train, and deploy machine-learning models in the cloud. Amazon SageMaker provides machine learning (ML) capabilities for data scientists and developers to prepare, build, train, and deploy high-quality ML models efficiently.
   (b) We created a Notebook instance in Amazon SageMaker as ml.t3.medium instance and named it as retail-sales-xgboost. It's an Amazon Linux 2 machine with Jupyter Lab 3 (notebook-al2-v2) application.
3. AWS Lambda
   (a) AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. A million requests are free in Free-tier.
   (b) We used AWS Lambda to connect the Machine Learning Model Endpoint with API Gateway to use the model as an API in a web application for retail sales prediction based on the input the user provides to the model.

4. Amazon API Gateway

   (a) Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. A million API Calls are free in Free-Tier program.

   (b) A REST API was created using Amazon API Gateway to communicate with the model endpoint to POST the data and GET the sales prediction from the model.

5. AWS Quicksight

   (a) Amazon QuickSight is a cloud-scale business intelligence (BI) service that you can use to deliver easy-to-understand insights to the people who you work with, wherever they are. Amazon QuickSight connects to your data in the cloud and combines data from many different sources. Amazon provides a 30-day free trial for new users. It includes 10GB of SPICE data storage which decreases the query time when creating dashboards and visualizations.

   (b) Amazon Quicksight was employed to visualize the dataset and understand the trends observed in the data. It provides

3.3 Application

1. AWS SageMaker

   (a) Data Loading and Preprocessing

      i. The Notebook instance retail-sales-xgboost was started. Using Jupyter Lab, we uploaded the dataset with all three files to Jupyter.

      ii. The files were read using Pandas Library in Python. The three files were merged into one data frame using the primary keys like Store, Date, and IsHoliday attributes.

      iii. As mentioned in the dataset, the Markdown columns 1 to 5 were added to only a part of the data in late 2011. The rest of the data in these columns contained only NA values. These values were removed to run the Machine Learning Model.

      iv. The IsHoliday column consisted of Boolean values of True or False. We encoded the values to 1 or 0 to pass it to the model.

   (b) Exploratory Data Analysis

      i. We used Matplotlib and Seaborn libraries in Python to visualize the data.

      ii. We observed that the Total weekly sales of different Store Types were not distributed uniformly. The type A store had the most weekly sales compared to the other two.

      iii. Using a Heatmap, we could observe that the Weekly sales were positively correlated with the Department and Size of the store. The larger the size of the store, the weekly sales for that store was higher. Similarly, some departments were performing well in weekly sales.

      iv. Also, we could see that the Markdown tends to increase with the IsHoliday value.

   (c) Machine Learning Model

      i. According to the Kaggle work by ENO5, *Time Series - ARIMA, DNN, XGBoost Comparison*[8] we concluded that the XGBoost Algorithm works best for time series data compared to other stat modes and Deep neural networks.

      ii. About XGBoost

         A. *XGBoost* [5] is a supervised learning algorithm and implements a gradient-boosted trees algorithm.

         B. The algorithm work by combining an ensemble of predictions from several weak models.

         C. Boosting is an ensemble machine-learning technique that works by training weak models in a sequential fashion.

         D. Boosting algorithms work by building a model from the training data, then the second model is built based on the mistakes (residuals) of the first model. The algorithm repeats until the maximum number of models have been created or until the model provides good predictions. *How it works* [6]

      iii. Advantages

         A. No need to perform any feature scaling.

         B. Can work well with missing data.

         C. Robust to outliers in the data.

         D. Can work well for both regression and classification.

         E. Computationally efficient and produce fast predictions.

         F. AWS can distribute the training process and data on many machines.

      iv. Disadvantages

         A. Poor extrapolation characteristics.

         B. Need extensive tuning.

         C. Slow training.

   (d) Model Training using Python

      i. We used the XGBoost library in Python to test the model performance on our dataset to predict the weekly sales with default parameters.

ii. Model Parameters
   A. The final input shape is (421570, 138)
   B. Output is Weekly sales as (421570,1).
   C. Training Data Size – 85
   D. Validation Data Size – 7.5
   E. Test Data Size – 7.5
   F. learning_rate = 0.1
   G. max_depth = 10
   H. n_estimators = 100

(e) Model Training using Amazon SageMaker
   i. Amazon SageMaker XGBoost model supports CSV and Libsvm as the input files. Here we used the uploaded dataset present in S3 bucket as the input to this model.
   ii. There are over 40 hyperparameters to tune Xgboost algorithm with AWS SageMaker.
   iii. We used Boto3 to connect to SageMaker session and pass the input files in S3 to the model.
   iv. We created a ml.m5.xlarge instance in Amazon SageMaker.
   v. Initial Hyper-Parameters
      A. The final input shape, output, training size, testing, and validation size remain the same as the Python model.
      B. learning_rate = 10
      C. colsample_bytree = 0.3
      D. eta = 0.1
      E. max_depth = 10
      F. num_round = 100

(f) Hyper-parameter Tuning
   i. *Amazon SageMaker allows us to run Hyperparameter Tuning Jobs* [7] on the model according to the dataset, which helps us find the best set of parameters to get maximum accuracy.
   ii. By utilizing this feature, we ran a hyperparameter tuning job on the model and found the following best parameters for our dataset.
   iii. Best Hyper-parameters
      A. Max_depth = 15
      B. colsample_bytree = 0.3913546819101119
      C. alpha = 1.0994354985124635
      D. eta = 0.23848185159806115
      E. num_round = 200
   iv. The model was then stored as a Model Endpoint in Amazon SageMaker in ml.t2.medium instance.

2. API Gateway

   (a) We developed a REST API in AWS API Gateway named retail-sales-forecast.

   (b) A POST Method was added to the API to send the user input data to the model endpoint.

   (c) A Stage was created as development to publish the API as a URL.

   (d) *API Link* is the invoke URL for the API.

3. AWS Lambda

   (a) A serverless function api-retail-sales-trigger was created in AWS Lambda using Python 3.9 environment.

   (b) This function calls the API and triggers the endpoint with the data received from the API using Boto3.

   (c) The function then sends a JSON output to the API containing the prediction results from the model.

4. AWS Quicksight

   (a) AWS Quicksight was used to create dashboards and visualizations on the dataset to observe any trends.

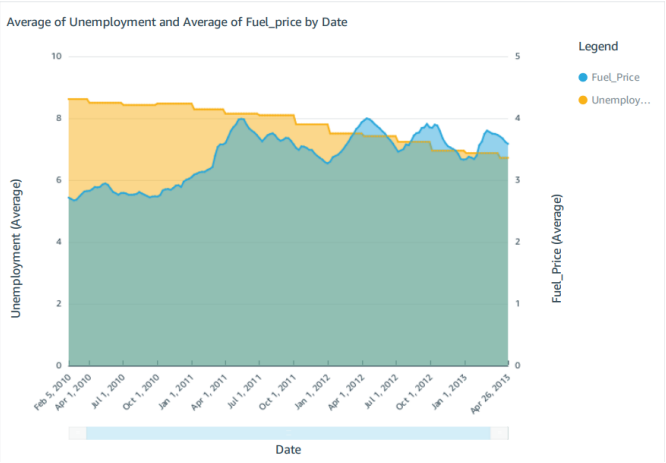

Figure 2. Average Unemployment and Fuel Price Vs Date

   (b) The above plot shows us the trend of average unemployment in the dataset with Average Fuel Price against the Date. We can observe that the fuel price is increasing gradually, whereas unemployment is reducing.

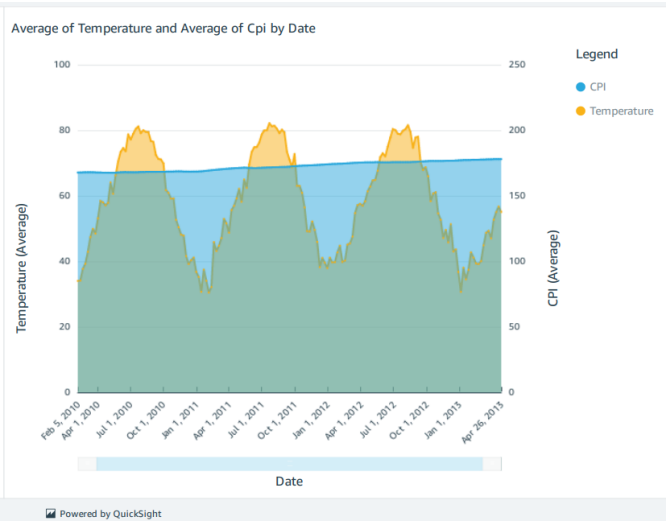

Figure 3. Average Temperature and Average CPI Vs Date

(c) The above plot shows us the average temperature variation and consumer price index against date. We can observe that during Summer, the temperature is higher and during winter, the temperature is the lowest. The CPI however is on a upward trend.

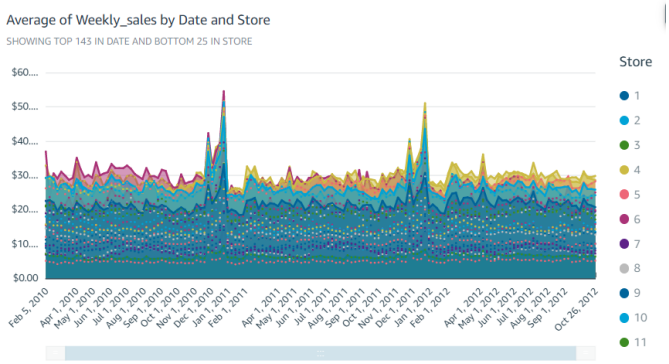

Figure 4. Average Weekly Sales Vs Date w.r.t Stores

(d) The above plot shows us the Average weekly sales trend against the date for every store. We can observe that during Holidays like Christmas and Thanksgiving, the weekly sales see a major jump.

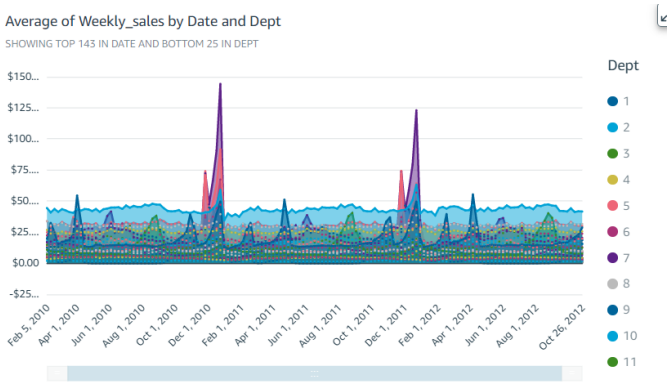

Figure 5. Average Weekly Sales Vs Date w.r.t Dept

.

(e) The above plot shows us the Average weekly sales trend against the date for every department. We can observe that during Holidays like Christmas and Thanksgiving, the weekly sales see a major jump, in almost all departments.

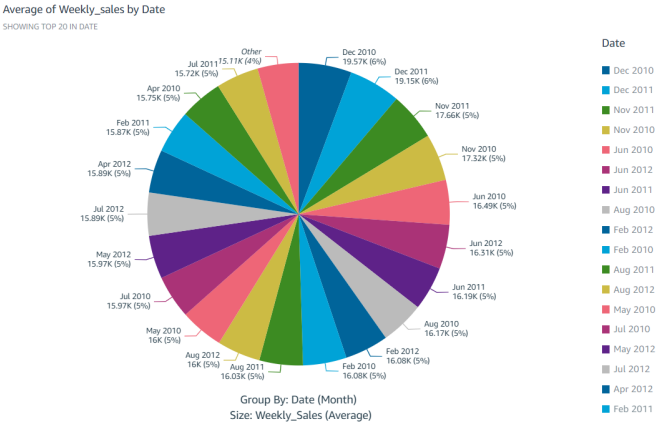

Figure 6. Average Weekly sales Vs Date

(f) The above plot visualizes the Average Weekly sales for every month in a pie chart. We can observe that the highest sales were during the month December 2010 and 2011.

5. Webpage Hosting

(a) A web application was developed using HTML, CSS and JavaScript

(b) The user inputs are present in a user-friendly form.

(c) Once the user submits input data, it posts the REST API with the data user has entered and returns the Predicted Sales Value as the Output which gets displayed on the webpage.

(d) Hosted the web Application publicly by using AWS S3 static website hosting service.

(e) Created an S3 public bucket named retail-sales-webpage and uploaded the web application in static website hosting and generated object URL and made it public.

(f) Retail Sales Prediction App URL

4. Results

1. Python XGBoost Model Results. We observed 90.5% accuracy on this model.

Table 1. XGBoost Accuracy

Metrics	Values
RMSE	7135.447
MSE	50914604.0
MAE	4010.5564
R2	0.905050155908306
Adjusted R2	0.9046339076639318

2. Amazon SageMaker Default XGBoost Model. We observed 89.8% accuracy on this model.

Table 2. Amazon SageMaker Default XGBoost Accuracy

Metrics	Values
RMSE	7141.338
MSE	50998708.0
MAE	4271.1924
R2	0.8986535169832639
Adjusted R2	0.8982092266736509

3. Amazon SageMaker Hyper-parameter Tuned XGBoost Model. We observed 98.2% accuracy on this model, which is the highest accuracy we got out of all the different models we tested with other hyperparameters.

Table 3. Amazon SageMaker Tuned XGBoost Accuracy

Metrics	Values
RMSE	3012.517
MSE	9075259.0
MAE	1215.1638
R2	0.9821232352545883
Adjusted R2	0.982044865753179

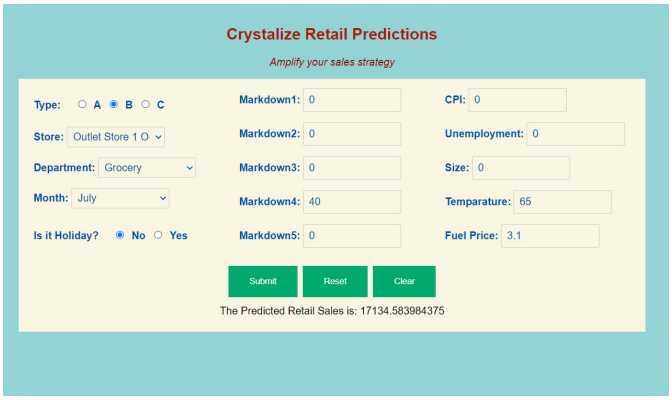

Figure 7. Webpage Layout and Result

4. The model is developed as a web application that can be used by any retail store to predict their retail sales depending on the previous week's sales and other parameters.

5. Future Work

1. Improve the Functionality of the web application to receive input from different sources dynamically.
2. Train the model with a larger and more sophisticated real-world dataset from any major retail store.
3. Incorporating feedback from end-users could help identify areas for improvement and guide future development of the application. This could involve user testing, surveys, or analyzing usage metrics to identify areas for improvement.

References

[1] *Dr. Bandaru Srinivasa Rao, Dr. Kamepalli Sujatha, Dr. Nannpaneni Chandra Sekhara Rao, Mr. T. Nagendra Kumar, "Retail Sales Prediction Using Machine Learning Algorithms" (2021)* URL

[2] *Devendra Swami, Alay Dilipbhai Shah, Subhrajeet K B Ray. "Predicting Future Sales of Retail Products using Machine Learning" arXiv preprint arXiv:2008.07779 (2020).*

[3] *Amazon Web Services. "Call an Amazon SageMaker model endpoint using Amazon API Gateway and AWS Lambda"* URL

[4] *Manjeet Singh. "Retail Data Analytics Dataset Kaggle"* URL

[5] *Tianqi Chen, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System" arXiv:1603.02754 (2016).*

[6] *Amazon Web Services. "How XGBoost Works"* URL

[7] *Amazon Web Services. "AWS SageMaker Hyperparameter Tuning"* URL

[8] *ENO5 Kaggle. "Time Series - ARIMA, DNN, XGBoost Comparison"* URL