Backorder Detection Using Machine Learning and Deep Learning Techniques.

Revision Number: 2.0
Last Date Of Revision 02/08/2021

Table OF Content

Overview

A backorder is an order for a good or service that cannot be filled at the current time due to a lack of available supply. The item may not be held in the company's available inventory but could still be in production, or the company may need to still manufacture more of the product.

Effect of Backorder

1. If a company consistently sees items in backorder, this could be taken as a signal that the company's operations are far too lean. It may also mean the company is losing out on business by not providing the products demanded by its customers. If a customer sees products on backorder—and notices this frequently—they may decide to cancel orders, forcing the company to issue refunds and readjust their books.

2. When an item is on backorder, a customer may look elsewhere for a substitute product, especially if the expected wait time until the product becomes available is long. This can provide an opportunity for once loyal customers to try other companies' products and potentially switch their loyalties. Difficulties with proper inventory management can lead to the eventual loss of market share as customers become frustrated with the company's lack of product availability.

Definitions

Term	Definition	Purpose
GCP	Google Cloud Platform.	Will deploy our machine learning model on this platform.
CloudRun	Google Cloud Serverless Container Engine.	Cloudrun will serve as our serverless container engine to host our model.
Container Registry	Will server as a place for storing model image	

Cloud Storage Bucket	Google cloud storage bucket.	Will store our model and retrieve it during prediction using python SDK.
Service account		This service account will allow cloudrun access to cloud storage bucket to access our model.

Problem Definition

After knowing what Backorder is and its disadvantages associated with it, we are going to build a machine learning model to predict if a product will be in the backorder or not.

Evaluation

The evaluation was to get a good accuracy score between 70 and 80 and a good precision and recall score between 60 and 85, then our model will be ready to be served in production.

Algorithms used

- Artificial Neural Network
- RandomForestClassifier
- DecisionTree
- NaiveBayes
- LogisticRegression
- K-Nearest Neighbour(KNN)
- Support Vector Machine(SVM)
- GradientBoostingClassifier

Evaluation Metrics

- Accuracy score
- F1 score
- Precision

- Recall
- Confusion matrix
- Classification report

Steps used to build model

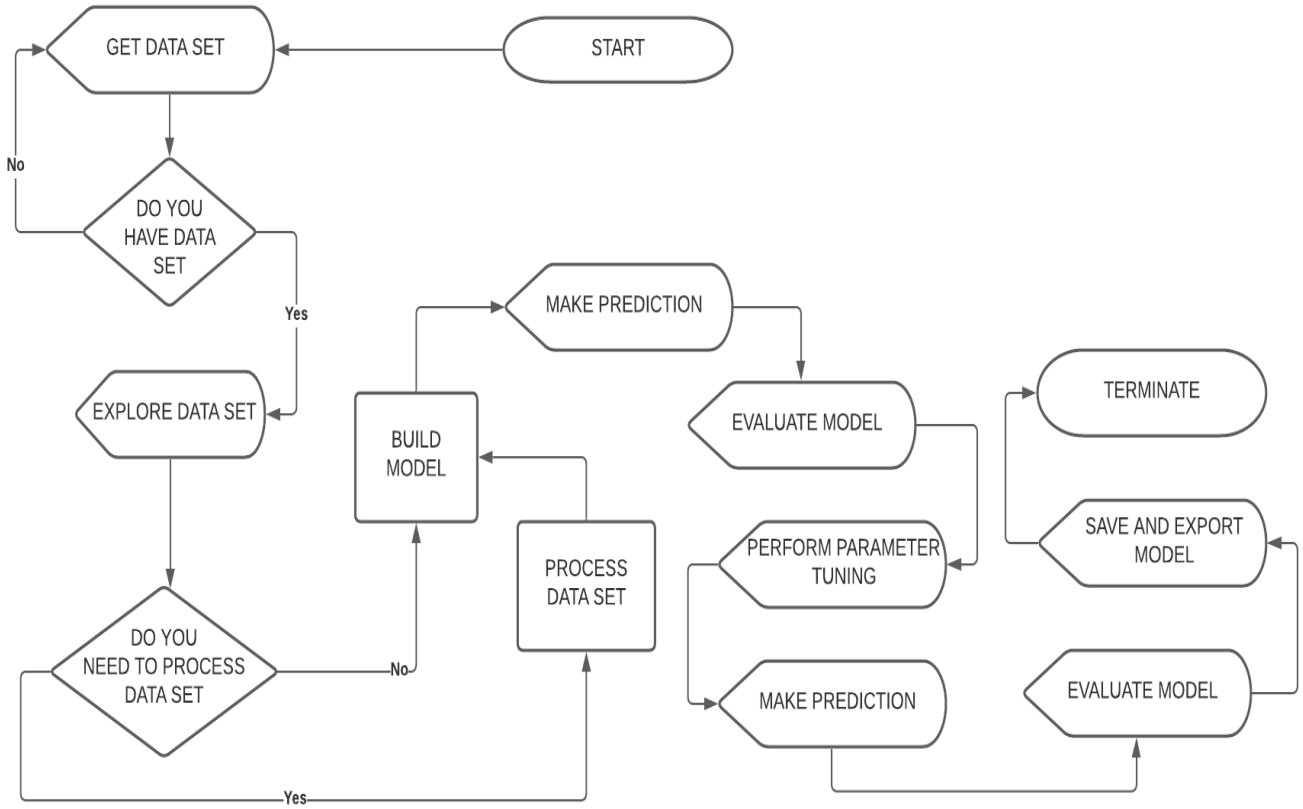
1. The model was built using different steps. The first step was to collect the datasets. The dataset was a secondary datasets, which was collected from kaggle.
Deployment
2. The second step was to explore the datasets. Here, activities such as checking null values, categorical and numeric features, outliers, highly correlated features, visualizing and many others.

3. The third step was to balance our dataset. Our dataset was not balanced. One class contained more values than the other hence we had to balance the datasets in order to have an equal number of values in each class.

4. The fourth step was to encode our label feature from category to numeric. After that we built our base model, made predictions and evaluated our model using a machine learning algorithm (Logistic Regression).

5. Eight different models were built until we chose the best model that had a good accuracy score, precision score, recall and f1 score.

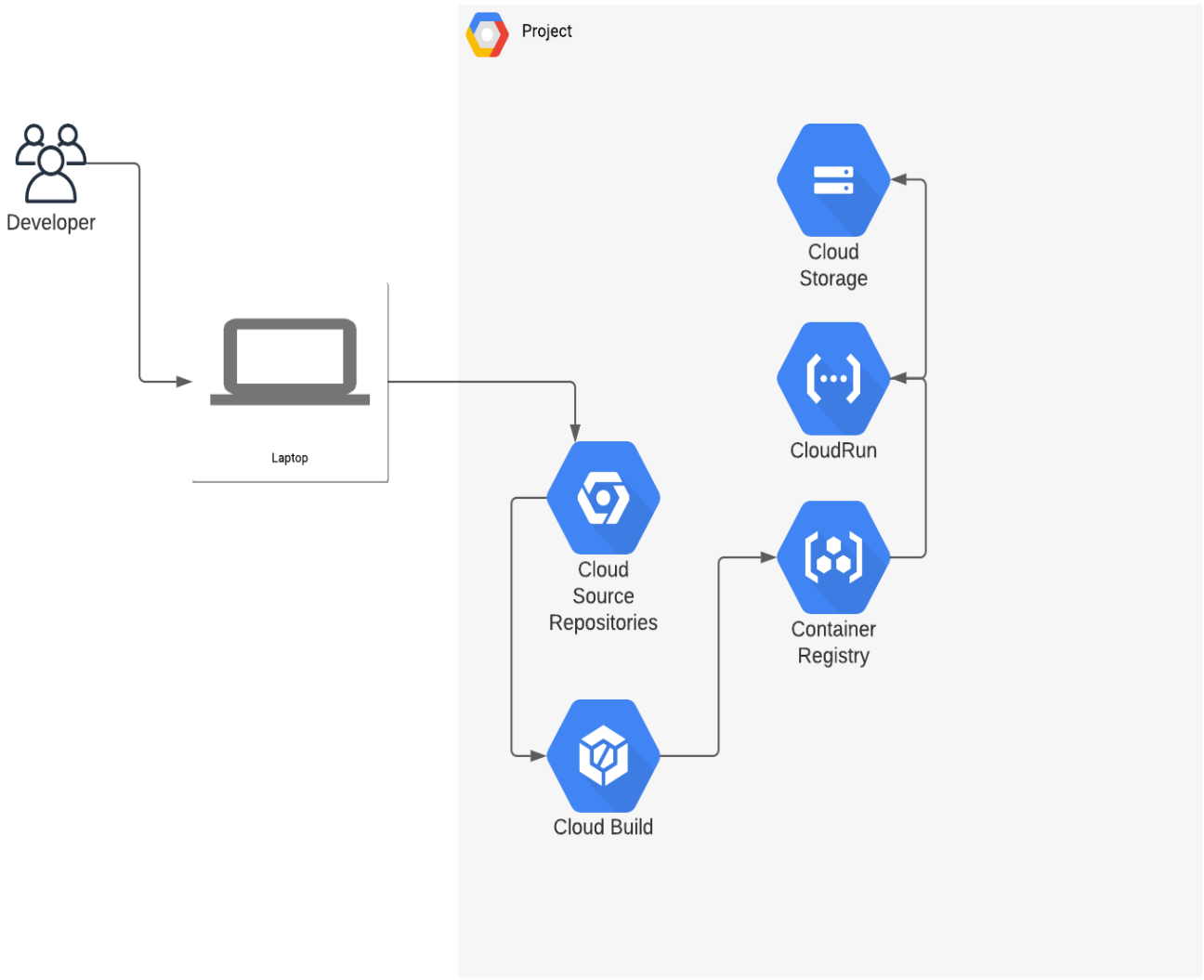
The diagram below Illustrate how the model was built



Deployment

The model was successfully deployed on Google Cloud Platform. The model was containerized and pushed to google container registry and deployed to google Serverless Container Services(CloudRun) using a script to automate the entire process.

The diagram below illustrates how the model was deployed on Google Cloud Platform.



Conclusion

The model can be deployed and be consumed by web applications, mobile applications and any other applications. Also the model can be retrained and redeployed using MLOPs techniques.