Forecasting demand for freight capacity by using Amazon SageMaker AI

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Forecasting demand for freight capacity by using Amazon SageMaker AI

Table of Contents

Forecasting demand for freight capacity by using Amazon SageMaker AI

Tianxia Jia and Hengzhi Chen, Amazon Web Services

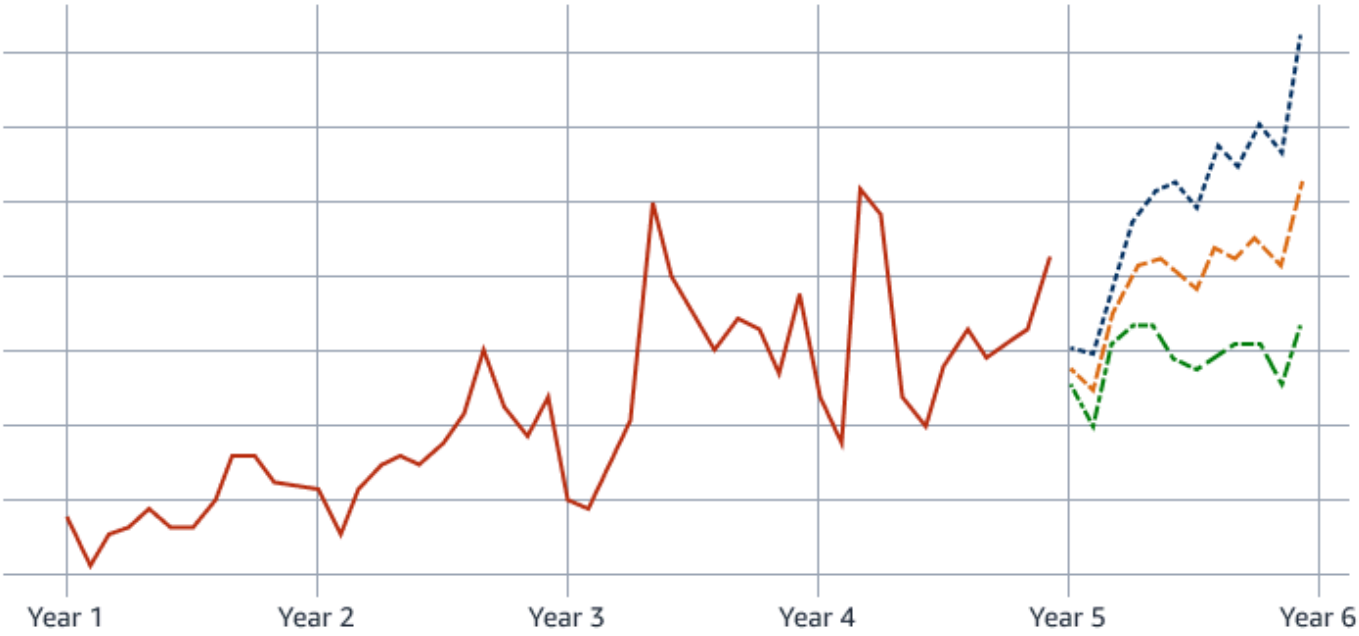
Demand forecasting is critical in the transportation and logistics industry, especially during periods of supply chain constraints. Accurate freight demand estimates benefit companies involved in logistics and supply chain, such as those that ship containers and air cargoes across borders. This helps companies effectively manage the cost of securing their transport network, which helps them manage shipping costs and maximize revenue and profit.

A machine learning (ML) model that is able to make an accurate forecast depends on high-quality training data. For demand forecasting, training data can include historical demand volume and other internally generated data that could be related to volume, such as price, inventory, and sales team headcount. In addition, external data, such as competitors, market environment, holidays, weather, and macro-economy, could also affect the demand volume. These internal and external data factors can be used as features in an ML model.

After all features are identified, the business may also want to provide input to some of these features that are within their control. For example, the business can set the shipping price in advance or decide when to do promotions and discounts. These types of user inputs can be incorporated into the model when making the forecast.

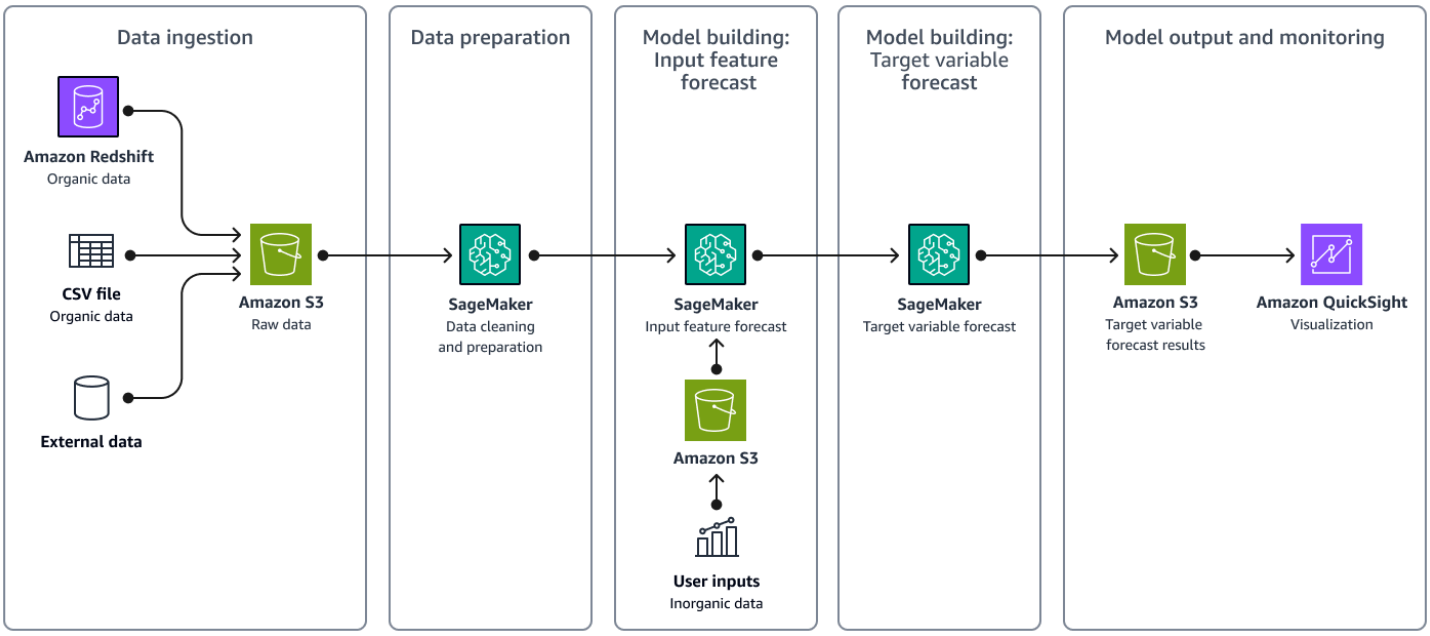
This guide describes a strategy to build a solution on AWS that makes an accurate logistic demand forecast by using an ML model. You train the model on a historical dataset that contains demand volume and features related to the demand. These features and metrics include both internal organic data and external data. The solution also provides the flexibility for the user and business analyst to provide inputs, which can then be incorporated into the forecast model.

The following image shows an example of a historical time series and 12-month forecasted range. You can use the recommendations in this guide to create an ML model that produces this type of forecast.


Year 1
Year 2
Year 3
Year 4
Year 5
Year 6

Architecture for forecasting freight demand

The following image shows the workflow of the solution, including data ingestion, data preparation, model building, and final output and monitoring.



The solution architecture includes the following main components:

1. **Data ingestion** – You store both organic data and external data in Amazon Simple Storage Service (Amazon S3).
2. **Data preparation** – Amazon SageMaker AI cleans up the data and prepares it for ML model training. For more information, see Prepare data in the SageMaker AI documentation.
3. **Model building: Input feature forecast** – SageMaker AI uses Prophet to generate a time series forecast for each input feature. You examine the forecast results. If needed, you provide user inputs to overwrite the feature's forecast.
4. **Model building: Target variable forecast** – SageMaker AI creates a regression model for inference by using the modified input features.
5. **Model output and monitoring** – The regression model outputs the forecast results to Amazon S3. You can visualize the forecast in Amazon Quick. Analysts can monitor the forecast results and evaluate accuracy by comparing the forecast with actual demand volume.

The entire processing pipeline from data ingestion to final model output can be orchestrated to run automatically. For example, you can set it up to automatically run monthly for a monthly demand forecast. If you need forecasts for more than one product, you can run the pipeline in parallel for multiple products. For more information, see Implement MLOps in the SageMaker AI documentation.

Data for forecasting freight demand

High-quality data is essential for any ML model to make a meaningful prediction and forecast. For demand forecasts, the dataset consists of any relevant data that could affect the ultimate demand. This data can come from various sources. You can classify this data into two categories, internal and external data.

Internal data

Internal data is organic, business-generated data. This data is usually stored in a data warehouse, such as Amazon Redshift.

You can directly generate or extract target output values from tables in the data warehouse that contain historical volumes for products of interest. For shipping companies, outputs or target values can be in units of full container loads for ocean shipping or total weight for air cargo.

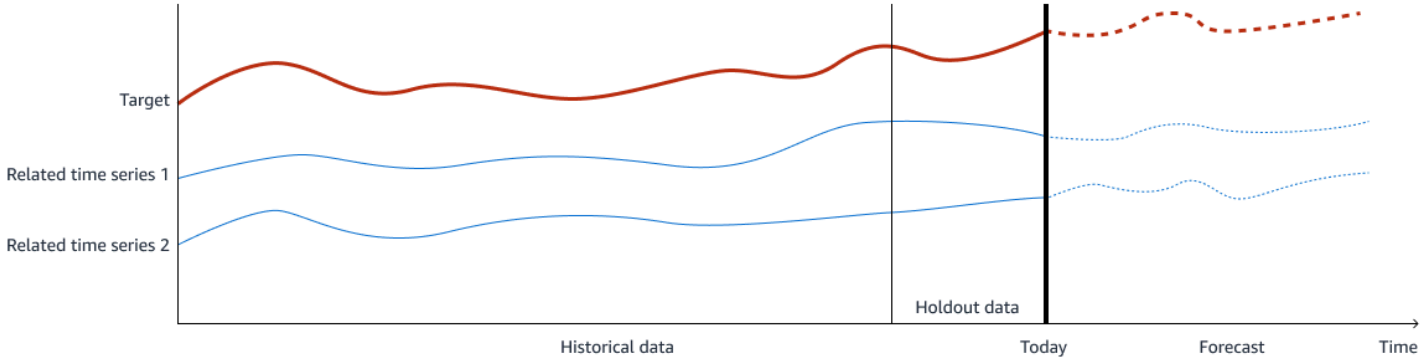
You can also generate various historical business metrics. These can be used as features in the machine learning model when forecasting demand. Example features include historical price, cost, capacity, and inventory.

External data

External data sources can be used as additional features to improve the forecast accuracy. Examples of external data sources include weather data, macro-economic data, industry data, and market data. These factors can have direct or indirect impact to the logistics and transportation industry, therefore affecting demand. For example, market freight rate provides a benchmark of the global freight market, which ultimately affects company-specific demand. Macro-economic data, such as import and export data for major economies, could also be used as a measure of market activity. To incorporate these external data sources, you can use various APIs to ingest data. For example, the St. Louis Fed provides the Federal Reserve Economic Data (FRED) API to access macro-economic data, and the National Oceanic and Atmospheric Administration (NOAA) provides the Climate Data Online (CDO) API to access worldwide weather data.

Machine learning models for forecasting freight demand

The following image shows an example of the training data. The target is what you want to predict, and related time series 1 and 2 are input features that are relevant to predict the target. Historical data is used for training and validation, and you withhold a period of the historical data for model validation.



In demand forecasting, the output (or target) is the demand volume that you want to predict. The input features are time series data related to the output. To train an ML model to make an accurate forecast of demand volume, two machine learning models are needed in the solution. The first model makes a time series forecast for the input features, including both internal and external data. The second model makes the final demand forecast by using all features. By using these two models together, you can effectively capture both the time series trend and the relationship between the target and the inputs.

ML model for the input feature forecast

Input features include both internal and external historical time series data. To make forecasts for each feature, you can use a one-dimensional (1D) time series model. There are various algorithms available. For example, Prophet works best with time series with strong seasonal effects and several seasons of historical data. A forecast is generated for each individual feature.

ML model for the target variable forecast

The ML model for the output, or the demand volume, is built to capture the relationship between all of the features and the output. You can use various supervised regression models, such as lasso, ridge regression, random forest, and XGBoost. When building the model and finding the best

parameters and hyperparameters, you can use holdout data. *Holdout data* is a portion of historical, labeled data that is withheld from the dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the predictions against the holdout data.

User inputs for forecasting freight demand

Although the futures of most of the features are unknown, there may be some features that the business has control over. For example, price is one feature that usually has a strong relationship with demand volume. Because the business sets prices, you know when prices will increase or when there will be a discount or promotion. In addition, the size of the sales team can also affect demand volume, and businesses control how big their sales teams are. For these data points that the business manages, you can provide user inputs. In the ML modeling step, the 1D time series models give a forecast for each feature, then users can examine these forecasted values and overwrite them with user inputs. These overwritten inputs are then used in the model when making the final forecast.

This user input step can be critical in situations where the 1D time series model forecasts do not match the business's expectations of the feature's future behavior. You can overwrite these forecasted values, which can improve the overall output forecast.

Best practices for an ML model that forecasts freight demand

By following these best practices, you can enhance the accuracy, reliability, and interpretability of your machine learning model for forecasting freight demand, ultimately leading to better decision-making and operational efficiency:

- **Data quality and preprocessing** – Make sure that the data used for training the model is of high quality and free from errors, missing values, and inconsistencies. Data preprocessing steps, such as handling missing values, outlier detection, and feature engineering, play a crucial role in improving the model's accuracy.
- **Sufficient historical data** – Having sufficient historical data is essential for capturing patterns, trends, and seasonality. However, it's also important to consider the relevance and timeliness of the historical data. If there have been significant changes in the market, business operations, or external factors, older data may not be representative of the current scenario. In this situation, give higher weightage to the more recent data.
- **Feature selection and engineering** – Identifying relevant features and engineering new features from existing data can significantly improve the model's performance. Collaborate closely with domain experts to use their knowledge and insights when selecting appropriate features. Additionally, consider performing feature importance analysis to identify the most influential features and potentially remove redundant or irrelevant features.
- **Ensemble models** – Instead of relying on a single model, consider using ensemble techniques that combine the predictions of multiple models. Ensemble models can outperform individual models and provide more robust and accurate forecasts.
- **Model evaluation and validation** – Regularly evaluate and validate the model's performance by using appropriate metrics, such as mean squared error (MSE), mean absolute percentage error (MAPE), or any other domain-specific metrics. Use cross-validation or holdout validation to assess the model's generalization capabilities.
- **Continuous monitoring and retraining** – Freight demand patterns can change over time due to various factors, such as economic conditions, market dynamics, or changes in business operations. Continuously monitor the model's performance and retrain it periodically with the latest data to improve its accuracy and relevance.
- **Explainable AI** – Demand forecasting models should be interpretable and explainable, especially in cases where stakeholders need to understand the reasoning behind the forecasts. Techniques

such as feature importance analysis, partial dependence plots, Shapley Additive explanations (SHAP) can help explain the model's decisions.

- **Incorporate domain knowledge** – Collaborate closely with domain experts and business stakeholders to incorporate their knowledge and insights into the modeling process. Their domain expertise can help identify potential biases, interpret results, and make informed decisions based on the forecasts.
- **Scenario analysis and what-if simulations** – Incorporate the ability to perform scenario analysis and what-if simulations into the forecasting solution. This allows stakeholders to explore the effect of different business decisions or external factors on the demand forecast, enabling more informed decision-making.
- **Automated and scalable pipeline** – Build an automated and scalable pipeline for data ingestion, preprocessing, model training, and deployment. This consistently and efficiently executes the forecasting process, especially when dealing with multiple products or regions.

Next steps

Before you implement this demand forecasting solution on AWS, it is recommended that you evaluate the problem you are trying to solve. It's a good idea to bring the business owners and data scientists together to brainstorm whether the problem can be solved by an ML model. It is critical to understand what datasets you have and the length of historical data available. It is also important for the business owners to collaborate with the data scientists to provide domain knowledge, identify useful features, and help create those features. The reliability of the model increases with the number of relevant features that you can create, which provides a more accurate forecast.

To build this architecture on AWS, begin by setting up an AWS account and provisioning the necessary services, such as Amazon Simple Storage Service (Amazon S3) for data storage and Amazon SageMaker AI for machine learning model training. Next, identify and collect the internal and external data sources that will be used as input features for the forecasting model. Store this data in Amazon S3 and use the data processing capabilities in SageMaker AI to preprocess and prepare the data for model training. In SageMaker AI, use automatic model tuning and distributed training capabilities to train and optimize the forecasting models. You can also use AWS services such as AWS Step Functions or AWS Lambda to set up a pipeline that periodically retrains the forecasting models with the latest data. After retraining, initiate a batch transform job in SageMaker AI to generate the forecast results, which you store in Amazon S3. Use Amazon Quick to visualize and monitor the forecast results generated from the batch transform job.

Resources

Amazon Quick documentation

- What is Amazon Quick?
- Working with data sources

Amazon SageMaker AI documentation

- What is Amazon SageMaker AI?
- Prepare data
- Perform automatic model tuning
- Use batch transform

AWS code samples

- Amazon SageMaker AI examples repository (GitHub)

Document history

The following table describes significant changes to this guide.

Change	Description	Date
Initial publication	—	May 14, 2024