Project Mid-Report

CS-579: Cloud Computing

Using Amazon Forecast to Build Forecasts – Comparative Analysis

Course Instructor:
Muhammad Usman Awais

Submitted By:
Ali Nauman – 18L1863
Hammad Akram – 18L1808

Project Overview:

This semester project is concerned around utilizing Amazon Forecast to forecast predictions for a foreseeable future. In this part, we shall be discussing our initial experimentation with Amazon Forecast utilizing *Electricity Usage* data-set. The number of evaluation metrics which are part of the Amazon Forecast. And how the data-set needs to be prepared for evaluation.

Using Amazon Forecast:

Amazon Forecast can be utilized through three ways,

- Console
- AWS CLI
- Python Notebook

In this project, we will be making use of Console functions and Python Notebook. For the initial understanding point of view, we need to see how this works.

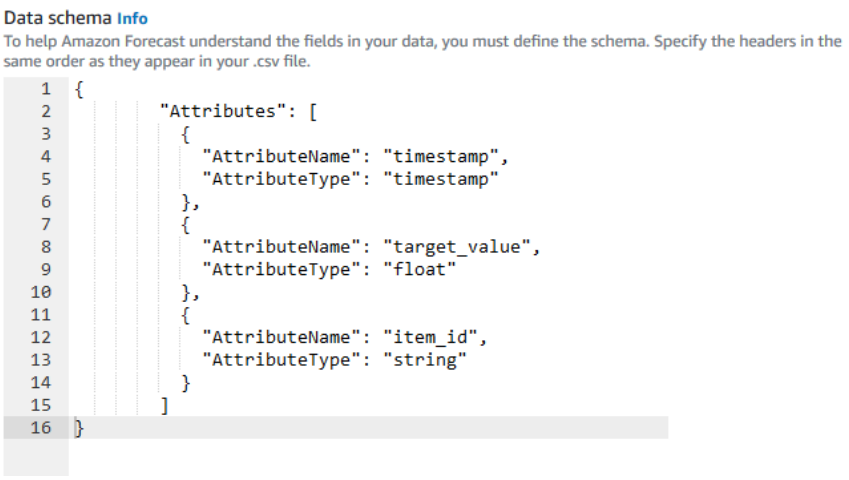
We need to import the data using another AWS Service, S3 Bucket. We will upload the pre-processed data into the S3 Bucket. From there, the Amazon Forecast Console, CLI or Notebook would access the data.

Firstly, we need to create a role to allow the access of S3 with Amazon Forecast. The details for this are stated in detail in the link below,

https://docs.aws.amazon.com/forecast/latest/dg/aws-forecast-iam-roles.html

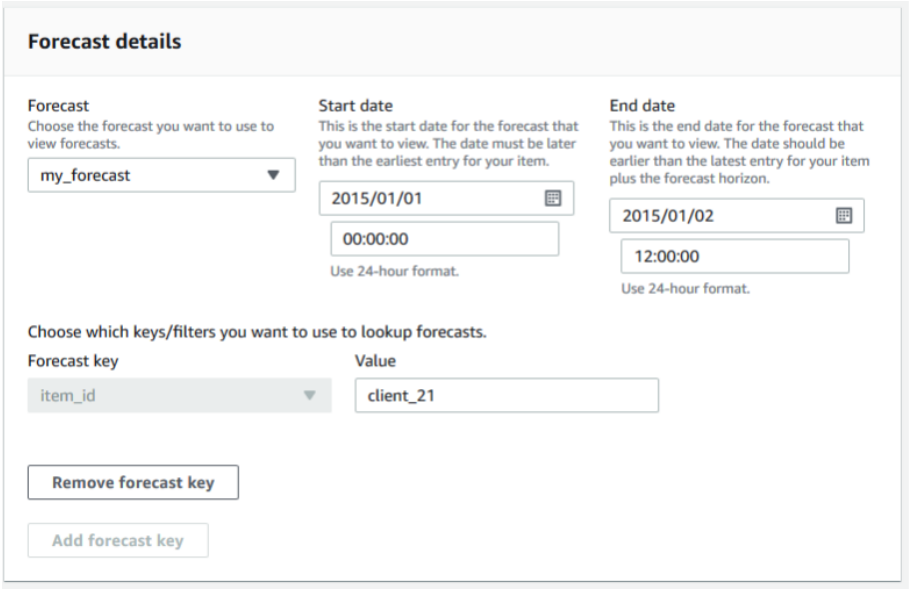
In pre-processing the data, we need to keep in mind that the timestamp column must have a standard or particular format suggested by Amazon i.e. yyyy-MM-dd or yyyy-MM-dd HH:mm:ss. While performing the pre-processing task, the timestamp can be adjusted accordingly.

Furthermore, we need to describe the data-set schema in JSON. This includes the features and the types of the features, as suggested below,

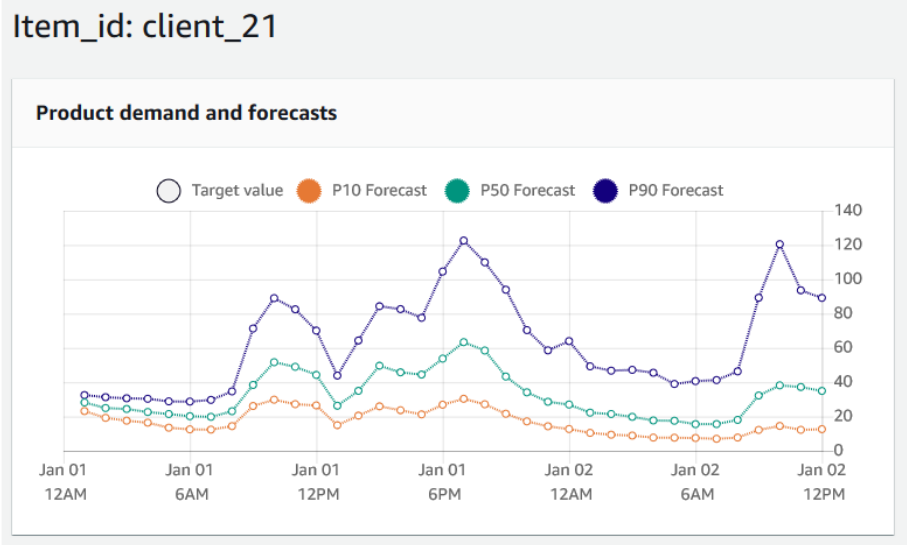


Next important task is to create a predictor. The predictor, which is a trained model, choose an algorithm and the number (length time's frequency) of predictions to make. You can choose a particular algorithm, or you can choose AutoML to have Amazon Forecast process your data and choose an algorithm to best suit your dataset group.

After this step, we need to create a forecast from the predictors created earlier. These inferences, will be used to create a forecast. A single forecast can be extracted through a query system, as it would be highlighted below. And we can retrieve the complete forecast for further use as well.



A sample result would look as follows,



Electricity Usage Data-set is a pre-processed data-set. The timestamp is in the correct format as suggested by Amazon, as shown below,

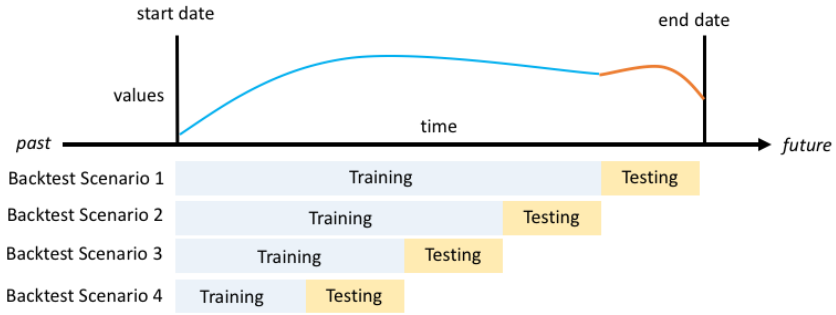
```
2014-01-01 01:00:00,2.53807106598985,client_0
2014-01-01 01:00:00,23.648648648648624,client_1
2014-01-01 01:00:00,0.0,client_2
2014-01-01 01:00:00,144.81707317073176,client_3
2014-01-01 01:00:00,75.0,client_4
2014-01-01 01:00:00,266.3690476190475,client_5
2014-01-01 01:00:00,6.359525155455055,client_6
2014-01-01 01:00:00,246.63299663299676,client_7
2014-01-01 01:00:00,50.69930069930072,client_8
2014-01-01 01:00:00,67.741935483871,client_9
2014-01-01 01:00:00,56.07302533532045,client_10
2014-01-01 01:00:00,177.12765957446825,client_11
2014-01-01 01:00:00,38.34991708126038,client_12
2014-01-01 01:00:00,36.0561582641991,client_13
```

It has a timestamp, usage, client columns. It has been used to forecast a 36-day forecast for every client for electricity usage. Using the above data-set, we have extracted a 9-day usage for a single client,

The thing to note here are the three values, P10, P50, and P90. These are the quantile predictors used. We can set our own predictors, or use the default standards, 10%, 50% and 90%.

Quantiles can provide an upper and lower bound for forecasts. For example, using the forecast types 0.1 (P10) and 0.9 (P90) provides a range of values known as an 80% confidence interval. The forecasted value at P10 is expected to be lower than the observed value 10% of the time, and the forecasted value at P90 is expected to be lower than the observed value 90% of the time. By generating forecasts at p10 and P90, you can expect the true value to fall between those bounds 80% of the time.

Including quantiles, Amazon Forecast has a Backtesting Parameter. It allows how many backtest windows to use to generate new forecasts. You can set both the backtest window length and the number of backtest scenarios when training a predictor.



Accuracy Metrics:

Amazon Forecast provides Root Mean Square Error (RMSE), Weighted Quantile Loss (wQL), and Weighted Absolute Percentage Error (WAPE) metrics to evaluate your predictors. Along with the overall metrics, Amazon Forecast also calculates the metrics for every backtest metrics.

Predictor metrics Info — CNN-QR

Evaluation type	Test window start	Test window end	Items	wQL[0.1]	wQL[0.25]	wQL[0.4]	wQL[0.65]	wQL[0.85]	WAPE	RMSE
Backtest average	-	-	0	0.1351	0.1578	0.2482	0.2623	0.2823	0.2828	276.8783
Backtest window 1	Tue, 07 Oct 2014 23:00:00 GMT	Fri, 31 Oct 2014 23:00:00 GMT	0	0.1351	0.1578	0.2482	0.2482	0.2482	0.2148	276.8783
Backtest window 2	Tue, 07 Oct 2014 23:00:00 GMT	Fri, 31 Oct 2014 23:00:00 GMT	0	0.1351	0.1578	0.2482	0.2482	0.2482	0.2148	276.8783

Weighted Quantile Loss (wQL):

The Weighted Quantile Loss (wQL) metric measures the accuracy of a model at a specified quantile. It is particularly useful when there are different costs for under-predicting and over-predicting.

Weighted Absolute Percentage Error (WAPE):

The Weighted Absolute Percentage Error (WAPE) is a commonly used metric to measure model accuracy. It measures the overall deviation of forecasted values from observed values. WAPE is always nonnegative, and a lower value indicates a more accurate model.

WAPE is more robust to outliers than Root Mean Square Error (RMSE) because it uses the absolute error instead of the squared error. For general use cases, it is the more informative metric.

Root Mean Square Error (RMSE):

Root Mean Square Error (RMSE) is a commonly used metric to measure model accuracy. Like WAPE, it measures the overall deviation of estimates from observed values. A lower value indicates a more accurate model.

Data-set Description:

In our project, we will be using the *Walmart Recruiting - Store Sales Forecasting* provided in Kaggle. It is the historical sales data for 45 Walmart stores located in different regions. Each store contains a number of departments, and every department includes a count of sales.

We are provided with a historical training data, which covers to 2010-02-05 to 2012-11-01. Within this file you will find the following details,

- Store – the store number
- Dept – the department number
- Date – the week
- Weekly_Sales – sales for the given department in the given store
- IsHoliday – whether the week is a special holiday week

The idea is to forecast this data for a particular number of days, and see how the values vary per the store and the department using Amazon Forecast.