Forecasting Sales of Consumer Devices Using Search Query Data

By Mayank Shrivastava[1], Konstantin Golyaev[2], Gagan Bansal[3], Matt Conners[4], Shahar Ronen[5], and Walter Sun[6]

Motivation

Most businesses require accurate revenue forecasts for efficient operations. This problem is front and center for manufacturers of consumer devices who face the need to coordinate supply chain operations over the course of months if not quarters. At the same time, forecasting sales of consumer devices, such as Xbox and Surface, is challenging for the business, partially due to lack of forecasting expertise, but also due to short sales history and strong holiday-driven seasonality.

To alleviate the challenges for our consumer device business, we turned to Bing query logs, anonymized and aggregated and processed by Bing Predicts algorithms. These days, people frequently do research on products before choosing what to purchase, e.g. comparing Xbox One against PlayStation 4 or Nintendo Wii U. Capturing and quantifying such research activities that typically precede purchases allowed us to construct much more accurate forecasts using machine learning techniques. For confidentiality purposes, we referred to devices as Device 1 and Device 2. We also referred to different geographic marketplaces as Geo 1, 2, and 3, for the same reason.

Approach

We discuss two separate approaches below, one that models weekly data (for Device 1) and aggregates weekly predictions to produce a quarterly forecast and the other that models monthly data (for Device 1 and Device 2) and aggregates monthly predictions to produce a quarterly forecast.

Weekly Models

In this approach, we used weekly historical data for Device 1 in Geo 1 to generate accurate quarterly forecasts. To ensure a single week was not split over multiple quarters, we adopted the following definition of week:

1) Week 1 of every year started from January 1st, and each week was seven days long, with few exceptions.
2) Each quarter had exactly thirteen weeks, so Week 13 could be six, seven, or eight days long.
3) During leap days, February 29th was assigned to Week 9, which made it eight days long.

[1] Mayank.Shrivastava@microsoft.com
[2] Konstantin.Golyaev@microsoft.com
[3] Gagan.Bansal@microsoft.com
[4] Matt.Conners@microsoft.com
[5] shahar@microsoft.com
[6] waltsun@microsoft.com

Using this definition, a one-quarter ahead forecast from a weekly model was obtained as follows. At the beginning of a quarter, we generated thirteen steps ahead forecasts at weekly granularity. These thirteen point forecasts were then summed up to yield the one quarter ahead forecast.

Features and Models

We employed two groups of features for our model:

1) Date-based features: week of year (1-52), month of year (1-12), quarter of year (1-4), year, and a linear trend.
    - Except for the trend, all other features are categorical.
2) Features from Bing Predicts: total weekly query counts grouped by intent: Retail, Review, Rumors, Support, Accessories, Comparison, Games, and How-To.
    - To avoid peeking into the future, we use date-based features and Elastic Net Regression to predict values of Bing features for the testing set, and use these predicted values in lieu of their actual values.

Our primary model was a Random Forest regression, referred to as "Best Model" in what follows. We also experimented with an Elastic Net regression, as well as Gradient Boosted Regression Trees, both of which demonstrated worse out-of-sample performance. As a baseline, we trained a Random Forest regression without any Bing features, i.e. with only deterministic date-based features (referred to as a "Simple Model"). We constructed forecasts for the past four quarters and we used the same model tuning procedure:

- Split data into training, validation, and testing sets.
    - The test set will be the quarter for which we construct forecasts (e.g. 2015-Q4).
    - The validation set will be the preceding quarter (e.g. 2015-Q3).
    - The training data will include all prior weeks (up to 2015-Q2, inclusive).
- We fit a number of models on the training dataset and use the validation set to tune hyperparameters. Then we combine training and validation data sets into one, retrain the model will the best hyperparameters, and construct predictions for the testing set.

We did not employ univariate time-series methods for these data. The ETS and ARIMA methods tend not to work well with weekly data. The STL method did work, but its performance was unimpressive, primarily because it does not accept additional features.

Results

In Table 1, we presented results from two models evaluated over different quarters of 2015, as well as their average performance.[7] For confidentiality reasons, we were unable to disclose actual forecasting errors, so instead we presented relative improvement metrics over a baseline forecast provided by the business. These numbers should be interpreted as follows: suppose that the mean absolute percentage error (MAPE) for the baseline forecast was *x* percent, and MAPE for the random forest regression

[7] The values in the "Average" row were computed by averaging the MAPEs over four quarters, and then transformed into relative values via the (x-y)/x function. As a result, 45 would *not* be the average of -12, 98, 100, and -354.

forecast was *y* percent. Then the numbers presented were computed as *(x-y)/x*. A larger number means more substantial improvement, and a negative number means the model did worse than the baseline.

Two major takeaways emerged from Table 1. First, on average even the Simple Model was able to improve over the baseline. Second, the Best Model performed much more consistently, and, in particular, it performed much better than the Simple Model in Q4 when lots of devices are sold during holiday season.

The random forest regression model also provided an ordered ranking of relative importance for the features used. The top three most important features from the Best Model were week of year, month of year, and number of Bing queries for Accessories.

Table 1. Device 1, Geo 1, Relative Improvements in MAPEs Over Baseline Forecasts

Quarter	Simple Model	Best Model
2015-Q1	-12%	93%
2015-Q2	98%	79%
2015-Q3	100%	94%
2015-Q4	-354%	79%
Average	45%	88%

Monthly Models

For Geos or Devices where sales data was not only available at a weekly granularity, we employed a monthly forecasting framework using monthly sales data. The primary goal was still to obtain an accurate quarterly sales forecasts. To arrive at quarterly forecasts, we generated three steps ahead forecasts at monthly granularity. These monthly forecasts were then aggregated up to obtain a one quarter ahead forecast.

Features and Models

We generated two sets of features for training our models:

1) Time series features based on sales data and seasonality:
    - Appropriately lagged actual sales (based on forecast horizon).
    - Month of year (1-12), Quarter of year (1-4).
    - A Boolean feature signaling if a new device was launched in a particular month.
    - Number of SKUs of the device being sold.
    - A Boolean feature signaling if the current month belonged to the holiday season.

2) Features from Web data:
    - Monthly query counts from Bing categorized by user intent for the device in question: Retail, Review, Rumors, Support, Accessories, Comparison, Games and How-To.
    - Monthly web page browsing counts from Internet Explorer, categorized by the classification of the visited domain: Retail, Review, Support, Games.

Complex models such as Random Forest regression or Gradient Boosted Regression Trees performed quite poorly while forecasting out-of-sample, because of a relatively small number of training samples at

a monthly granularity, owing, in turn, to the short histories of the devices in question. We employed the following process to train ensembles of linear regressions to generate the predictions:

- All the months prior to a given month were used while generating predictions. The evaluation was done on a "rolling" test set, with each new month getting added to the training set to generate predictions.
- We trained a large number of linear regression models, each by randomly selecting a few features from the two feature sets described above, while ensuring that the total number of features per model remained under a fixed threshold.
- We ranked the performance of all linear regressions over the past six months using root mean squared errors (RMSE). Models in the top 10 percent were chosen to form an ensemble. The final predictions were generated by taking the median of the individual predictions of each model in the ensemble.
- To generate forecasts for different horizons, the features based on Web data were lagged appropriately to generate features for models which predict a month, two months and three months out.

Results

In Table 2, we presented results from our models for Device 1 evaluated over the four quarters of 2015 and Geo 2 and Geo 3, with the metric being used is the relative improvement in performance over the baseline. The metrics were computed in the same way as in Table 1.

Table 2. Device 1, Geos 2 and 3, Relative Improvements in MAPEs Over Baseline Forecasts

Quarter	Device 1 Geo 2	Device 1 Geo 3
2015-Q1	97%	97%
2015-Q2	-19%	95%
2015-Q3	57%	-148%
2015-Q4	29%	14%
Average	53%	68%

In Table 3, we presented analogous numbers for our models for Device 2, evaluated during the same time period across all three Geos.

We also generated a measure of relative feature importance by measuring the proportion of models, selected as part of the ensembles, having a non-negligible coefficient for a particular feature. The most important features, for models for both Devices 1 and 2 measured using this approach were date-based features (month, year), seasonal features (holiday period, new device release) and query counts for Accessories and Reviews.

Table 3. Device 2, Geos 1, 2, and 3, Relative Improvements in MAPEs Over Baseline Forecasts

Quarter	Device 2 Geo 1	Device 2 Geo 3
2015-Q1	9%	-11%
2015-Q2	-6103%[8]	59%
2015-Q3	61%	76%
2015-Q4	69%	21%
Average	44%	46%

Conclusion

Accurate revenue forecasts are critical for efficient operations of the consumer hardware devices business. By using consumer intent features engineered from Bing query logs as part of a machine learning workflow, we were able to improve the mean absolute percentage errors for quarterly revenue forecasts by anywhere between 44 and 88 percent relative to the baseline forecast provided by the business.

[8] The -6103% number looks intimidating, but it is primarily an artefact of an extremely accurate baseline for this quarter.