Whova Sign-up Rate Analysis

Yumeng(Ophelia) Wang

October 24, 2022

1 Exploratory Data Analysis

The dataset has the number of attendees and the number of sign-uped users for each event in January 2020 and in January 2022. To analyze the pattern of the sign-up rate, I first visualized the sign-up rate time series using the event start-date as the timestamp for each event. (Figure 1).

Figure 1: Sign-up rate on the start-date (2020 January and 2022 January)

Since multiple events can have same start-date, the plot above shows the minimum, average and maximum of sign-up rates on each day. If we would like to analyze the average of sign-up rate as a time series, we could use ARIMA model to model that. However, since the dataset given only has two-month data which is also disjoint, the sensibility of using time series analysis is not very strong. If I have a continuous time series data, I would first model a deterministic linear function of time and then use ARIMA model to model the sensibility as well as predicting the white noise.

Hence, I decided to do a event-based study to predict the sign-up rate for different kinds of events. Below are the histograms of sign-up rate in 2020 January and 2022 January. (Figure 2).

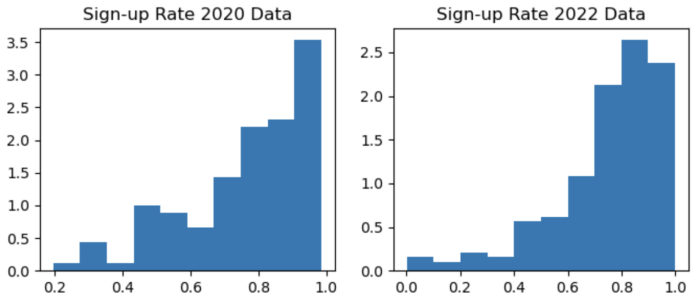

Figure 2: Sign-up rate on the start-date (2020 January and 2022 January)

The histograms of 2020 and 2022 data do not show significant difference. However, if we split events by event type, either in-person, virtual or hybrid, they show quite different distributions of sign-up rate. (Figure 3). In the dataset, we have 162 in-person events, 112 virtual events and 35 hybrid events.

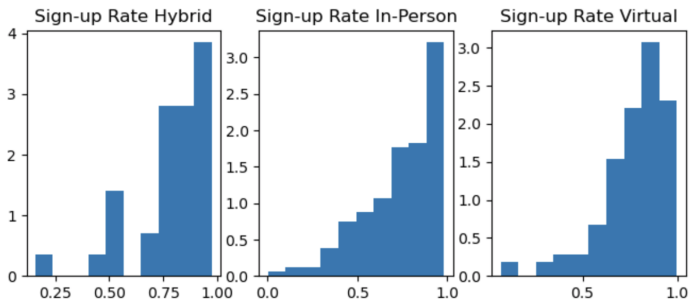

Figure 3: Sign-up rate based on event type

Event Type	min	25%	50%	75%	max
In-Person	0.002463	0.628963	0.787079	0.904987	0.984375
Virtual	0.066667	0.707407	0.812084	0.892431	0.998254
Hybrid	0.157658	0.733931	0.848387	0.921124	0.974138

There are some features that we can observe from Figure 3:

- In-person events has the longest left tail. The risk of having extremely low sign-rate is higher if we hold a in-person event.
- Hybrid events seem to have a better overall performance in terms of having the highest 25%, 50% and 75% quantiles.

2 Event-based Study - Feature Engineering

If we would like to predict the sign-up rate based on events, we need to discover sensible and meaningful features for build the prediction model.

2.1 Dataset Provided Features

There are some features provided by the dataset:

- Event Start Date: the start date of the event.
- Event End Date: the end date of the event.
- Event Size: the total number of attendees of the event.
- Event Type: could be one of the following: in-person, virtual, hybrid.

2.2 Self-designed Features

There are some features I designed based on the information provided by the dataset:

- Duration: the number of days from the start date to end date of the event
- Business Days: the number of business days from the start date to end date of the event, excluding US holidays and weekends. It also reflects the number of holidays/weekends if we include the duration feature into the model.
- Business Days Proportion: the ration of business days to duration.
- Start Date Weekday: the weekday of the start day.
- End Date Weekday: the weekday of the end day.
- Start Date Holiday: boolean to indicate whether the start date is a holiday or weekend.
- End Date Holiday: boolean to indicate whether the end date is a holiday or weekend.
- Is 2020: boolean to indicate whether the event is in 2020 or 2022.

3 Modeling the Sign-up Rate - Classification

At first I tried to fit a regression model to predict the sign-up rate, achieving RMSE with 0.3. How good the regression model is depends on the MSE or RMSE. However, the criterion of small MSE is subject to discussion as the magnitude of MSE/RMSE depends on the magnitude of the data. To better interpret the model, I change the regression model into a classification model by discretizing the sign-up rate into multiple categories.

Classification Label	0	1	2	3	4	5
Sign-up Rate	(0, 0.5]	(0.5, 0.6]	(0.6, 0.7]	(0.7, 0.8]	(0.8, 0.9]	(0.9, 1]

The figure below shows the independent and dependent variables in the model. More details are included in the feature engineering subsection.(Figure 4).

	duration	business_days	start_weekdate	end_weekdate	start_holiday	end_holiday	buisiness_days_prop	event_type	is_2020	size	rate_category
0	5	2	4	1	False	False	0.400000	0	1	525	2
1	3	1	4	6	False	False	0.333333	0	1	236	3
2	4	2	3	6	False	False	0.500000	0	1	396	5
3	3	1	4	6	False	False	0.333333	0	1	127	0
4	3	2	3	5	False	False	0.666667	0	1	283	5
...	...	...	...	...	...	...	...	...	...	...	...
314	3	2	3	5	False	False	0.666667	0	0	88	0
315	2	1	2	3	False	False	0.500000	1	0	498	5
316	6	3	3	1	False	False	0.500000	1	0	266	5
317	3	2	2	4	False	False	0.666667	2	0	1024	4
318	4	2	6	2	False	False	0.500000	0	0	224	4

309 rows × 11 columns

Figure 4: Independent and Dependent Variables

I splitted the data into training set and test set by a ratio of 3:1 and fitted different classficaton models.

3.1 K Nearest Neighbor with Cross-Validation

I first used the KNN model to fit the sign-up rate categorical labels. By tuning the value of k, the number of neighbors, using cross-validation, the optimal k is 44 and the model achieved training accuracy 0.43% and test accuracy 0.44%.

3.2 Decision Tree with Cross-Validation

Then I fitted the model with decision trees with cross-validation and tuning on parameters such as max_depth, min_samples_leaf, min_weight_fraction_leaf and max_leaf_nodes. The best grid returns a decision tree model with train accuracy 0.67% and test accuracy 0.67%.

3.3 Random Forest with Cross-Validation

Lastly I fitted the model with random forest with cross-validation and tuning on parameters such as bootstrap, max_depth, max_features, min_samples_leaf, min_samples_split and n_estimators. The best grid returns a decision tree model with train accuracy 0.77% and test accuracy 0.59%. The random forest model achieves the highest test accuracy because of bagging and boosting, however it achieves a slightly lower test accuracy than the decision tree even with cross-validation to lower the model complexity.

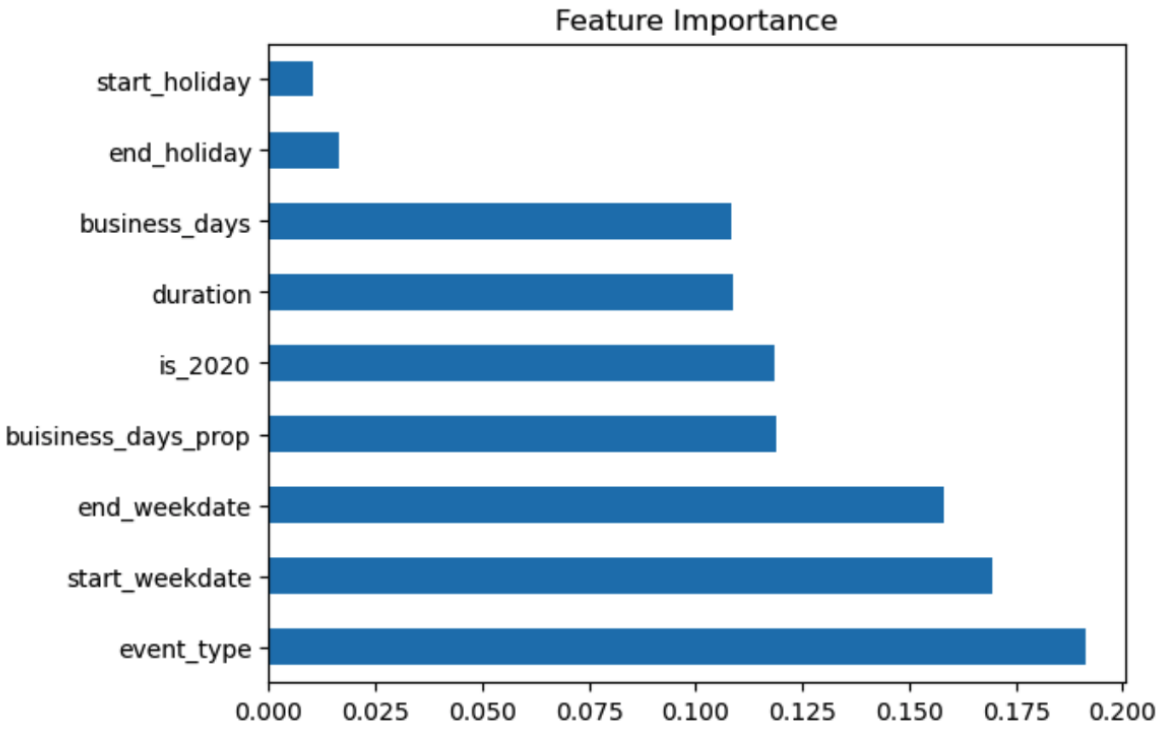

Figure 5: Feature Importance from Random Forest Model

The figure above shows the feature importance from the best random forest model I tuned. (Figure 5). From the plot, we can see that features such as event_type, start_weekday, end_weekday, business_days_prop and duration are the most predict features in the model.

The figure below is the visualization of one tree included in the random forest model, showing the slitting criterion and classification results. (Figure 6).

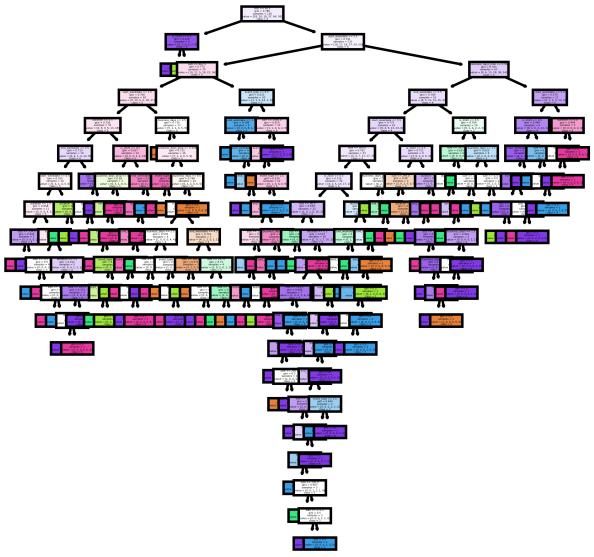

Figure 6: Feature Importance from Random Forest Model

Overall I think the amount of features we can build from the dataset is quite limited. The majority of the features I built comes from the perspective of the event type and event time. If I can have access to more data such as the company holding the event, how Whova promoted the app during the event, the activity flow of the event, I can build more predictive features based on the event.

4 Time Series Analysis

If I have data with a longer time window, I will do a time series analysis of the average sign-up rate of events held on the same day, using the end date as the timestamp. This is not a very good method to use given the data, but below is a try with the 2022 virtual events data.

4.1 Time Series Visualization

Below is the visualization of the average sign-up rate for virtual events in 2022 January(Figure 7). In total we have 22 data points. 16 of the data points are used as the training set.

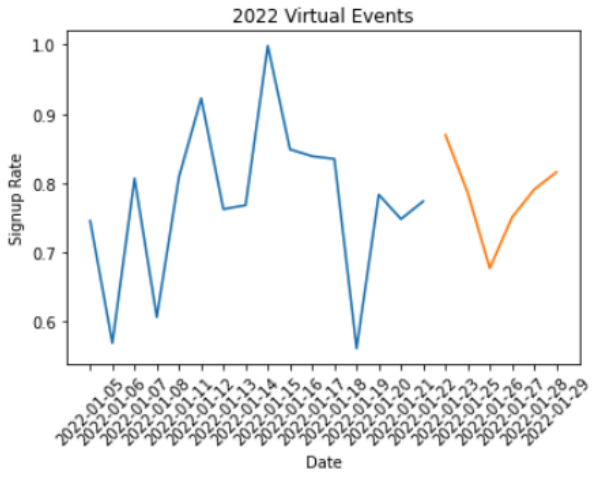

Figure 7: Time Series of Average Sign-up Rate Train & Test - 2022 Virtual Events

4.2 Stationary Test

Usually I will remove the linear trend in the time series first by running a linear regression against time. However, due the limited data we have, there's no significant linear trend I can observe, so I go ahead testing the time series for stationary using one of the unit root tests, the Augmented Dickey-Fuller Test.

```
from statsmodels.tsa.stattools import adfuller
ts = ts_0['rate']
result = adfuller(ts)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
```

```
ADF Statistic: -17.782739
p-value: 0.000000
Critical Values:
        1%: -4.138
        5%: -3.155
        10%: -2.714
```

Figure 8: Test of Stationarity with Augmented Dickey-Fuller Test

The test outputs significant p-values under different p-value cutoffs, showing the time series is stationary and therefore ready for fitting into a ARIMA model.

4.3 ACF Plot

I first examined the ACF plot of the time series. If the ACF plot shows significant spikes at any lag order, it can help me to better identify the parameters inside the multiplicative seasonal ARIMA model.

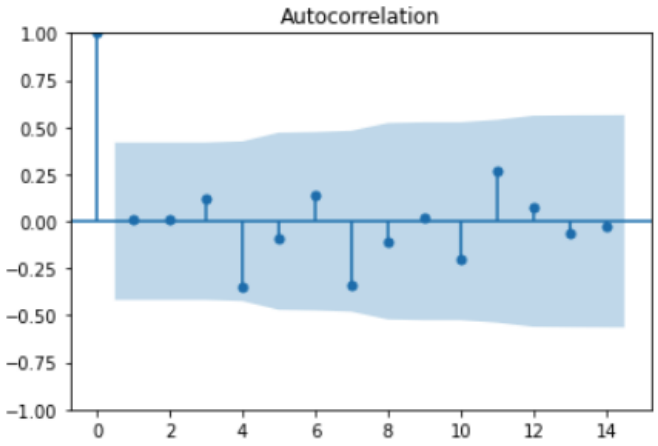

Figure 9: Test of Stationarity with Augmented Dickey-Fuller Test

The plot above doesn't show any significant spike at any lag order. Hence, the time series of average sign-up rate is already a white noise (Figure 9).

4.4 Fitting MSARIMA Model

Before reach the conclusion, I still try to fit the multiplicative seasonal ARIMA model with paramters tunning. The performance metric I used here is AIC.

Model	AIC
ARIMA(0,0,0)(0,0,0)	-20.288
ARIMA(1,0,0)(0,0,0)	-18.288
ARIMA(0,0,1)(0,0,0)	-18.288
ARIMA(1,0,1)(0,0,0)	-16.305

We prefer model with lowest AIC. Hence the best model here is ARIMA(0,0,0)(0,0,0), which is consistent with my previous finding that the time series is already a white noise. It's hard to make predictions here because either it is completely random or we just don't have enough data points.

5 Conclusion

I analyzed the pattern of sign-up rate from two perspectives: a event-based study that predicts the sign-up rate for an event based on more than 10 features of the event, and a time-series analysis of the average sign-up rate on each date. The event-based prediction achieves descent classification accuracy. However it definitely can be improved if I have more data about the event. The time-series analysis shows that either I don't have enough data points since I can only analyze January 2020 or January 2020 data or the average sign-up rate is completely random, with very minimal predictive power.