Airline Tweets Sentiment Forecast
Zihao Guo

Data

The raw data contain 14640 instances, 14 features and 1 target ‘airline_sentiment’ (negative, neutral, positive). Detailed explanation is in notebook. There are mostly negative sentiments, which makes sense because people tend to post comments online when they are not satisfied with something. Our goal is to predict the likelihood of negative tweets in the future across all airlines even though differences among airlines exist. We are mainly interested in what time makes negative sentiment more possible.

Data Visualization

Referring to notebook, we can see American, US Airways, United have REALLY unbalanced sentiments (many neg). Virgin America has most balanced sentiments but doesn't have many records since it's a relatively small airline compared to others. Here, one possible direction for research is to figure out why different airlines have such big different ratio of 3 sentiments. We then move to figure out the negative reasons for negative sentiment. We can identify that bad customer service and late/canceled flights account for over 70% negative sentiments. Since we want to predict the future negative tweets, time those tweets posted is crucial. Unfortunately, we only have a week tweets available for us to explore. Normally, we hope for several months or years data so that we can forecast the following week/month negative tweets based on prior week/month. Thus, we can only predict the following day based on prior day, which might not work well because more people travel on Sunday and Monday.

Data Engineering

We won’t perform this sentiment analysis as a NLP task. Instead, we’ll mainly work with time series model embedding machine learning algorithm, so I make a lot of changes to our feature space. First, I drop 3 variables with over 90% missing values. Then, I fill NaN with -1 for positive sentiment in the ‘negativereason_confidence’. I further delete suffix ‘-0008’ in ‘tweet_created’ because it’s the same for all records. Next, I create dummy variables for nominal variables, such as ‘airline’, ‘negativereason’. Finally, I change ‘negative’ to 1, ‘neutral’ to 2, and ‘positive’ to 3 in ‘airline_sentiment’. There are also some small variations to the data which are shown in notebook.

After data transformation, I look at the mutual information of features with target. Unsurprisingly, ‘negative_reason_confidence’ has the largest mutual information with `airline_sentiment`, but we know it's kinda cheating since it's another measure of sentiment and we don't know the algorithm to compute it. However, since we have this variable in our raw data, we might want to use it for better model performance.

Since we only have 7-day tweets but 14640 instances, we want to bin our data for every 10 mins and we can predict the next day sentiment based on the prior day. Thus, we have a data of shape (7,144,104), where 7 means 7 days, 144 means 144 10-min windows each day, and 104 is the number of features after data transformation. Even though our data show high negative sentiment on Sunday and Monday, we can still train our model by puting Sunday into train and Monday to test. Furthermore, we want full day records, so we'll abandon data on 02/16 and 02/23, which are incomplete (no whole day records). Note: our problem here becomes a regression problem, because airline_sentiment now is the mean of sentiments in each bin.

Models

We know machine learning algorithm predicts a single value for each instance, but we want a multi-step forecast model with multiple variables. Thus, we want to develop a framework that can incorporate machine learning into multi-step time series forecasting. This can be achieved by either using 'airline_sentiment' as both input and output, or using all features to run ML model and then forecast future sentiments using time series.
We use RMSE as our evaluation metric, which measures the difference between predicted mean value of sentiment and real mean value of sentiment in each bin. We then use walk-forward evaluation to forecast step by step. This means we want a prediction of a day, then the real data for that day is provided as input so that it can be used as basis for predicting the subsequent day.
I pick 6 models for ML algorithm. First 3 are linear algorithms, including 'linear regression', 'lasso', 'ridge'. Then, I pick Random Forest, SVM for regression and XGBoost to examine ensemble methods and gradient boosting algorithm. Linear models work well if the feature space has linear relationship with target. However, if the internal relation is not linear, we might want to use an ensemble method, which combines several weak learners to form a stronger learners by averaging. Furthermore, among ensemble methods, XGBoost, in general, performs better than random forest or SVM, because it fits a new model to the residues of the previous prediction rather than assigning new weights to predictors. This loss-targeted improvement makes XGBoost a really popular ML algorithm.

Experiment & Conclusion

I deploy two strategies of multi-step forecasting.
Firstly, I only use airline_sentiment as input and output to run time series forecast. This configuration yields a 0.291 RMSE in XGBoost algorithm, which is pretty optimal, considering the only thing we use for prediction is sentiment.
Secondly, I use all variables to run forecasting. However, the result is less optimal. There are several reasons. First, since we have a really sparse feature space, our models don't assign much weight to those variables, so further data engineering is required (reduce dimension). Second, In the first strategy, the model uses 6*24 = 144 windows to predict next window, but this case uses 144*104 = 14976 windows to predict the next window, making weights for each previous window' parameters are extremely low, so model can't predict well. Third, I only use default setting for ML models, performance of which might be greatly improved using Cross-Validation. Thus, second strategy is not necessarily worse than the first one.
Therefore, we achieve a RMSE 0.291 using default setting of XGBoost. Thus, for a new window(10 minutes) on a new day, we can use the time series model to capture the mean sentiment of tweets in that window, so we have a likelihood of sentiment in terms of values from 1 to 3. Furthermore, if we just want the number of negative tweets in that window, we can filter out positive and neutral sentiments at the beginning and run time series model on the number of negative sentiment rather than the mean sentiment.

Future Work

1. More data engineering! Dimension reduction
2. More data! Tweets only for a week are not enough for comprehensive analysis
3. Tune parameters for each model!
4. Deploy more complex deep learning models!
5. Analyze sentiments for **each airline** to give it advice on improving its service and customer satisfaction.!