The Application of TVAR model on Financial Data

Wenwen Ye
Statistical Department of Duke University

Time-Varying Spectral Analysis

Introduction

This project aims to conduct time-varying spectral analysis with WTI crude oil price data, and explore the hidden cyclic structure of time series data, by applying the time-varying autoregression model and analyzing it on the spectral domain. Given the potential seasonal pattern that the data might have, we can also predict the how long it takes to get to the next maximum.

Methods

1. Time-varying auto-regression model[1]:
   $y_t = x_t + v_t$
   $x_t = \Sigma_{i=1}^{p} \phi_{t,i} x_{t-i} + \epsilon_t$ where $\epsilon_t \sim N(\mu, \sigma)$
2. Use Forward-filtering backward-sampling technique and obtain a full trajectory of $p(\phi_t, v_t|D_T)$, and make posterior inference on ϕ
3. Plug in posterior median of ϕ_t and obtain the spectral density for auto-regression functions AR(p) at each time point t:
   $f_t(\omega) = \frac{v_t}{2\pi|1-\phi_{t,1}e^{-i\omega}-\cdots-\phi_{t,p}e^{-ip\omega}|^2}$ [1]

Data

WTI crude oil price data 2014-08-08 to 2018-11-05. The data is downloaded from https://fred.stlouisfed.org/series/DCOILWTICO/. It is a time series data with 2306 observations. To correctly calculate spectral density at each time point, we need to transform data into a stationary time series.

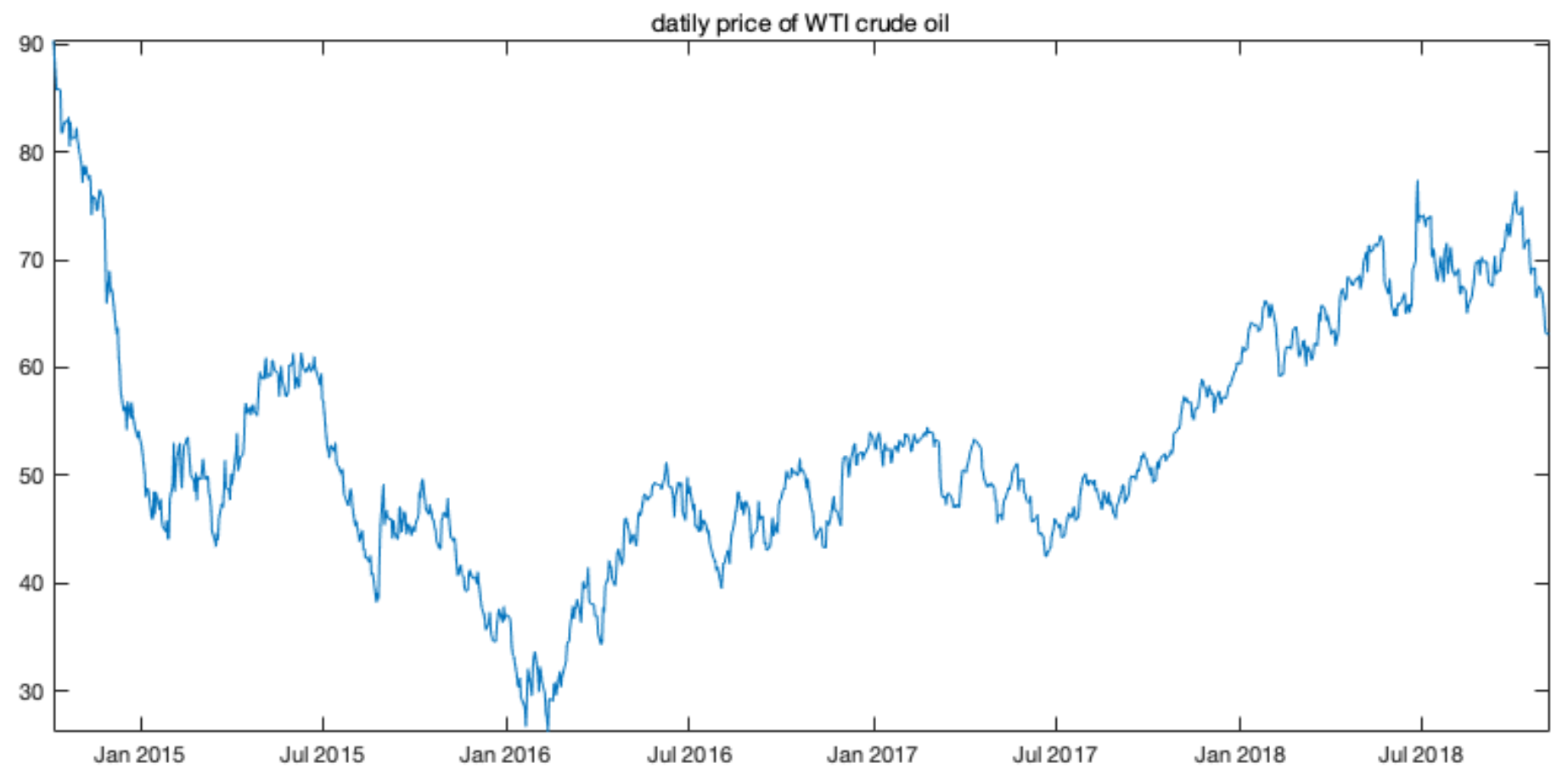

Fig. 1: WTI Oil daily price

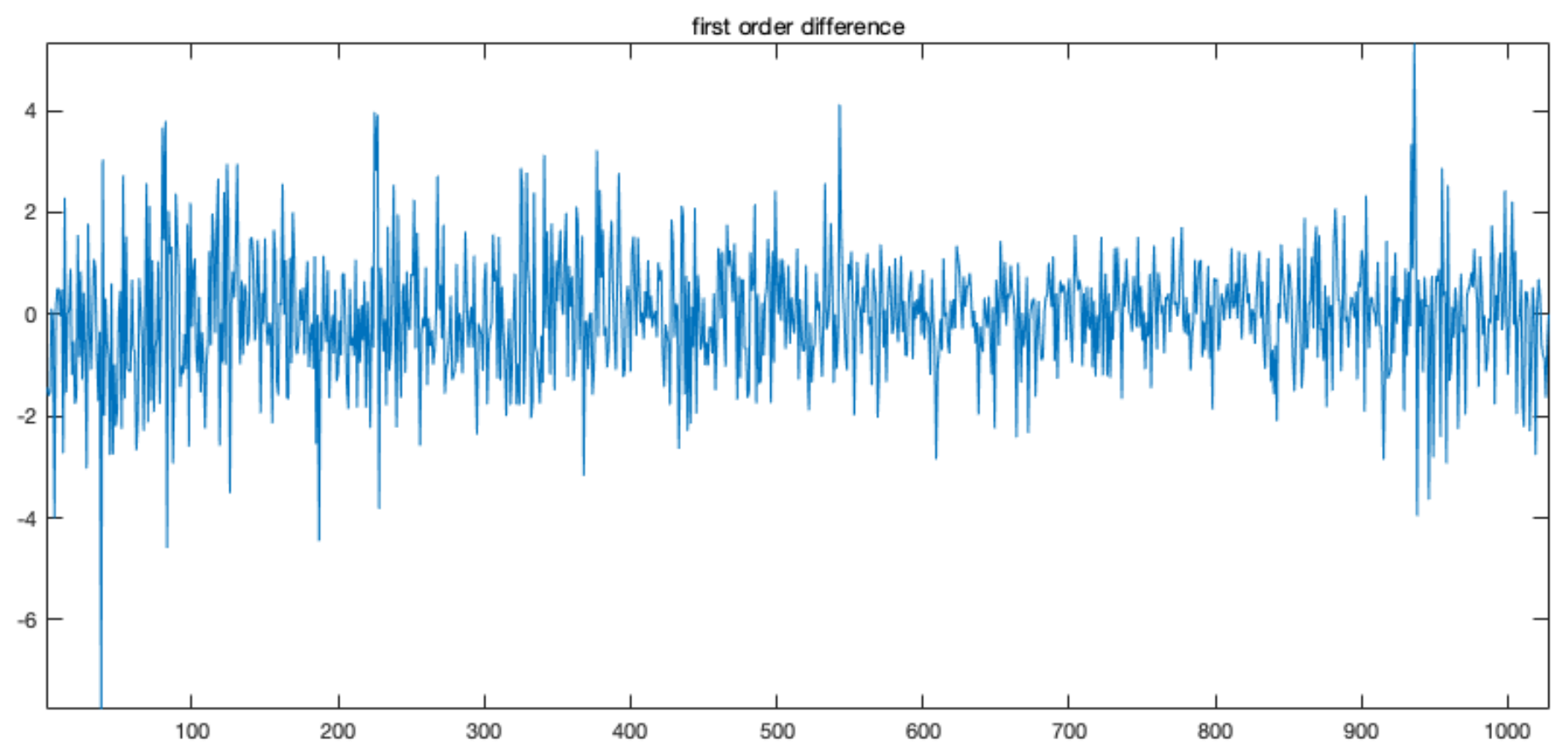

Fig. 2: first order difference

Conclusion and Prediction

1. Peaks of spectrum at most of the time points are around -0.54, from which the potential cyclical periods is calculated as $\frac{2\pi}{\omega} \approx 11$ days.

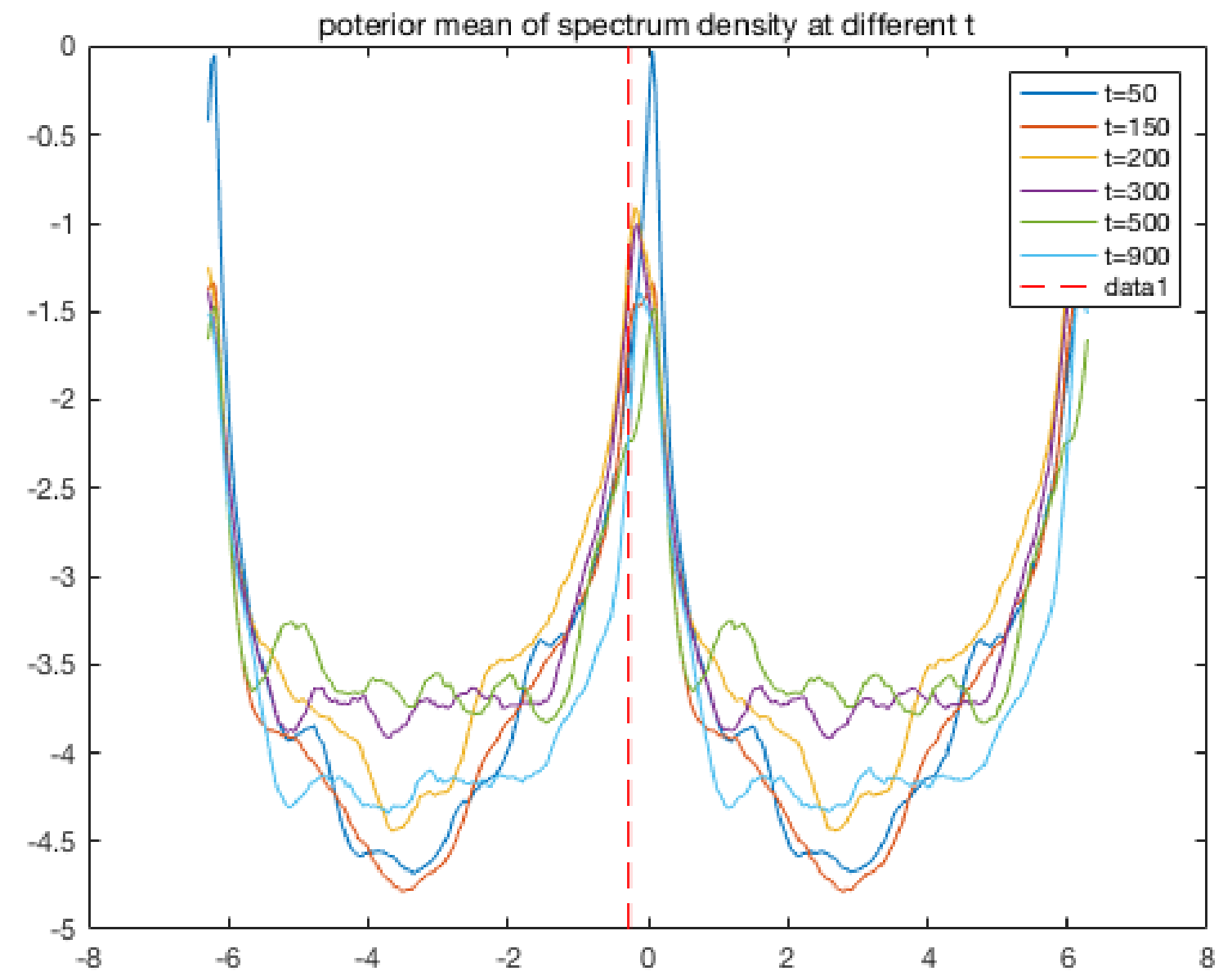

Fig. 3: spectrum at different ts

2. Simulate 1000 samples with MCMC and get a distribution of times to next maximum
3. Obtain the density function of frequency at each time point and can therefore knows when the prices are more likely to be in a trend versus in volatile status.
4. Corresponding trading strategy: only take action when prices are in trend

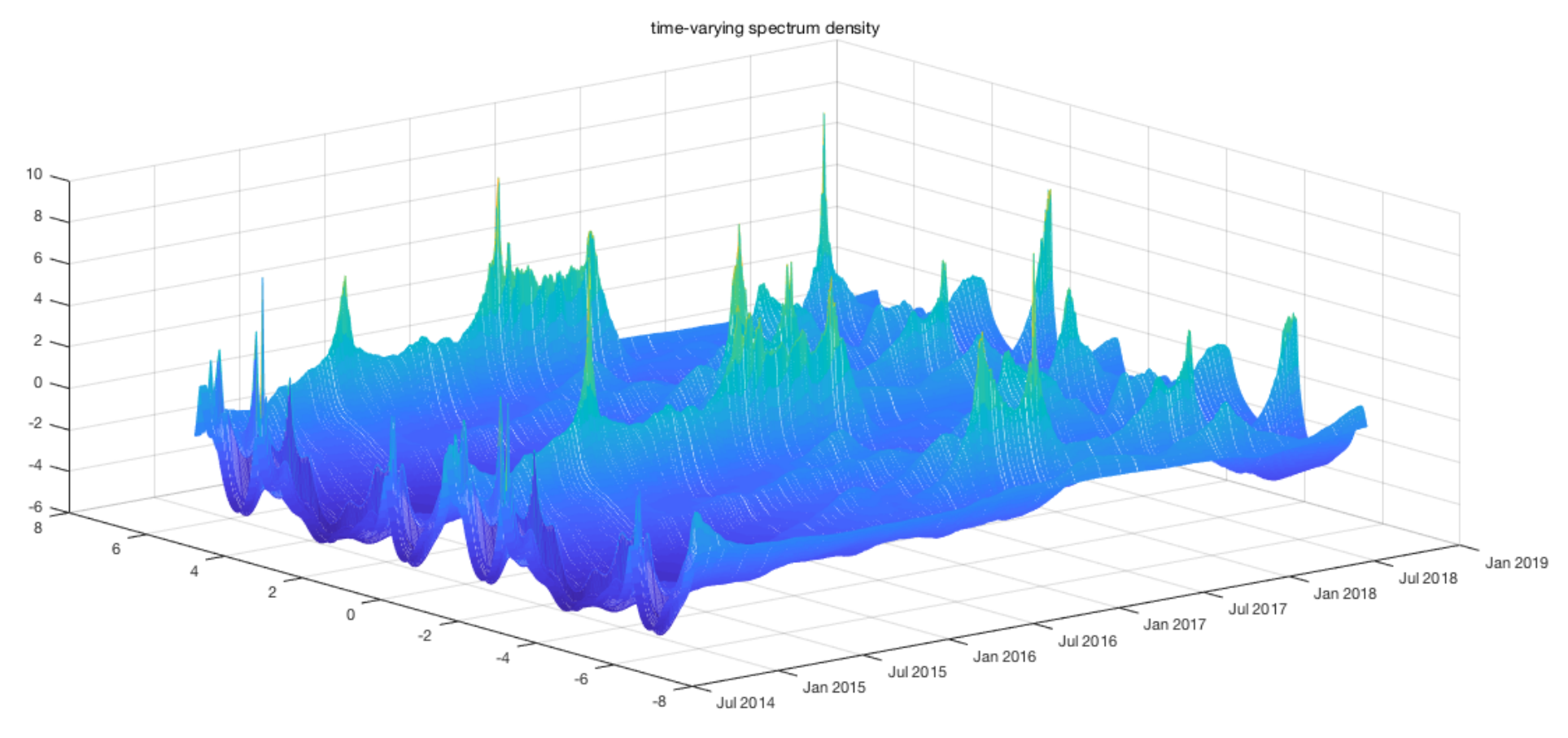

Fig. 4: spectral density at every time point

Discussion

1. frequency peak very close to zero
2. qqplot of residual show deviations from a straight line at extreme values

How Long Does a StackOverFlow Question Closing Take?

Introduction

This project aims to predict whether a StackOverFlow question will be closed within 1 hour, given various features about the question, and users that answers the question. This will be helpful for an user to estimate how long he/she will get a descent answer from responders. This project has mainy three parts: extracting the data from Kaggle.com[2] with Google BigQuery API, selecting the useful features for predictions, and applying machine-learning techniques with the selected features.

Data

1. Total of 178GB data size and can only be queried and analyzed partially
2. Classify the closing time into several time intervals, and the majority of all questions are closed within a very short time.

closed within			
1h	**5h**	**10h**	**24h**
55.6%	73.7%	78.7%	85.7%

3. There are a few helpful features in the orginal dataset. Therefore we created a dummy variable for each of the 50 most frequent tags.
4. The feature we select to predict labels are as follow:

title	title of the question
score	score of the question
view_count	number of views
ans_score	score of the best answer
creation_date	time when the question is created
tags	topic of the question

Methods

1. Gradient Boosting: XGBoost in scikit-learn package
2. Feature selections picks out the tags that are most helpful for classifications:
3. Used BayesianOptimization function in bayes_opt package to tune parameters in gradient boosting models and drastically improved the tuning time comparing to a grid-search approach.

Conclusion and Discussion

The classifier that predicts whether the question will be closed within one hour achieves 68.3% accuracy score, and roc curve with 0.69 AUC:

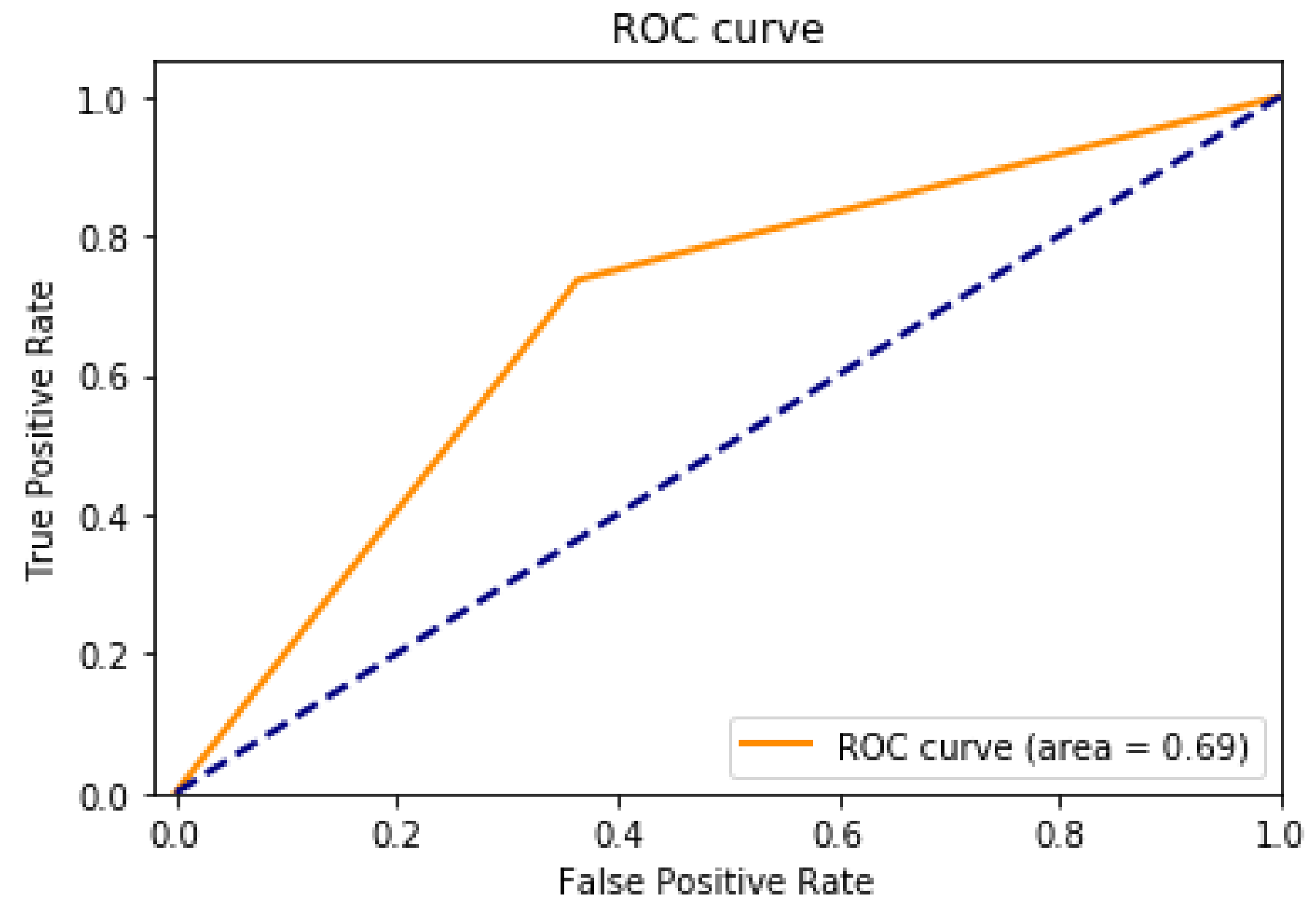

Fig. 5: ROC curve

Potential Future Improvement:

1. Alternative ways of transforming response variable, such as exponential values.
2. Text mining of each question's title or body

Acknowledgement

Cooperated with Kai Wang, Junwen Huang, and Lianghui Li

References

1 Prado, Raquel, et al. Time Series: Modeling, Computation, and Inference. Chapman Hall/CRC, 2018.
2 StackOverflow. “Stack Overflow Data.” Kaggle, 12 Feb. 2019,