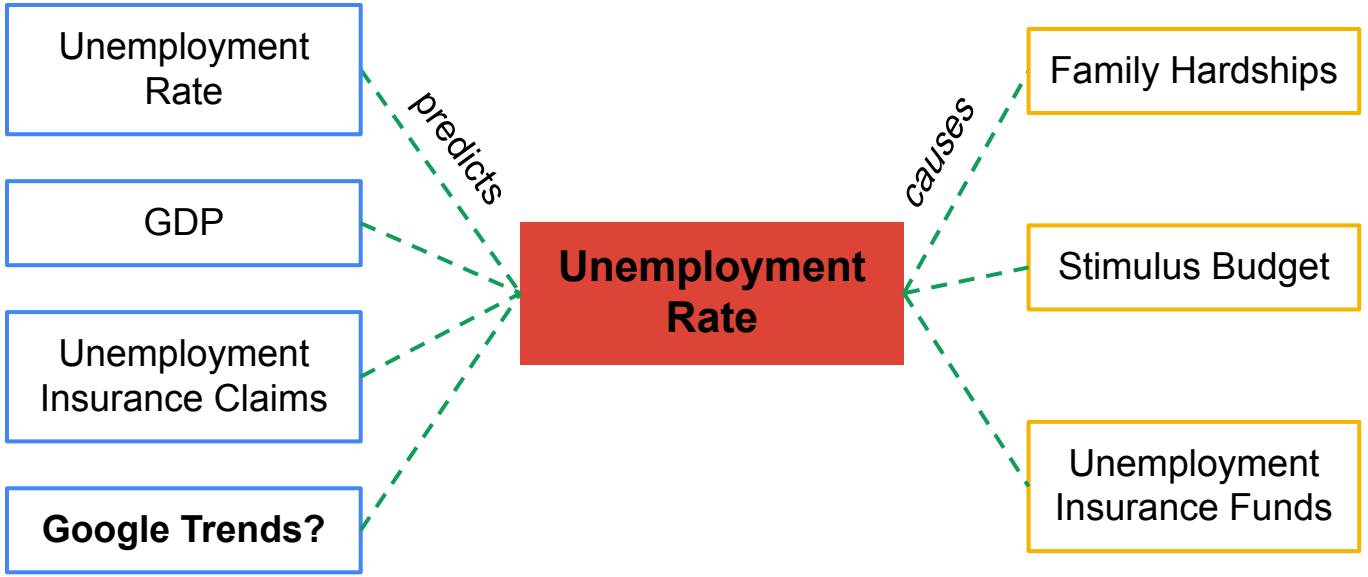
Predicting Unemployment Rate with Google Trends Keywords

Group 10: King of Data

Research Problem



Data Collection

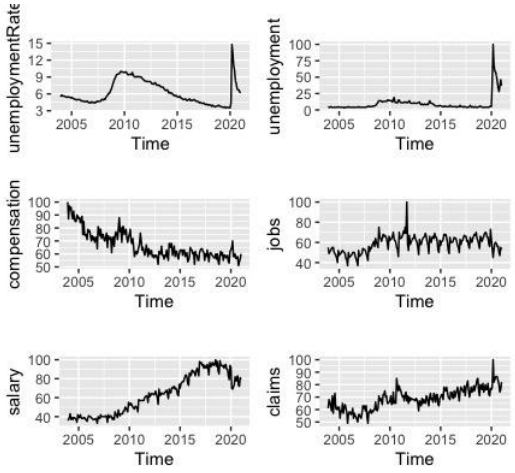
Google Search Term Frequency

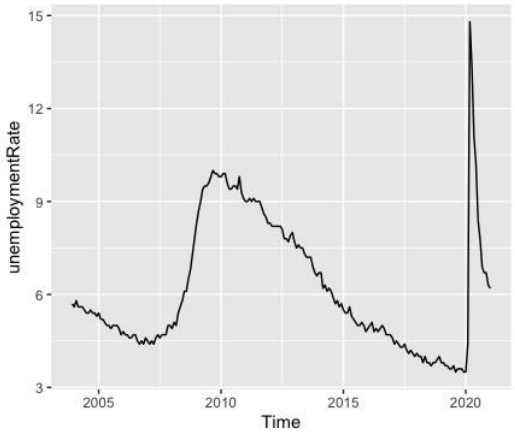
- 2003 ~ present
- “Unemployment”
- “Compensation”
- “Jobs”
- “Salary”
- “Claims”

Unemployment Rate

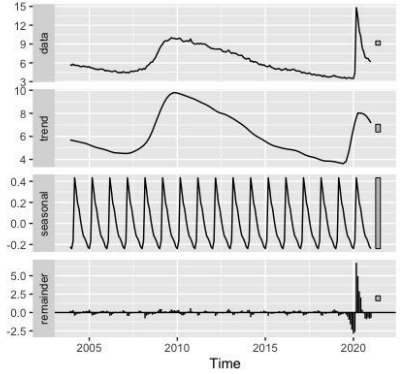
- 1948 ~ present

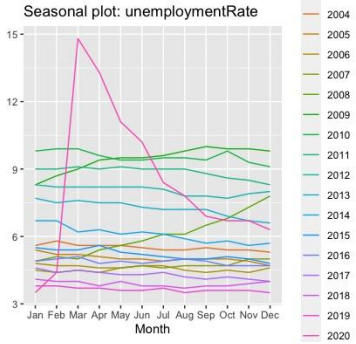
```
> head(data)
  X       date unemployment compensation jobs salary claims unemploymentRate
1 1 2003-12-31            5          100   55     36     61              5.7
2 2 2004-01-31            4           87   49     37     68              5.6
3 3 2004-02-29            4           97   51     39     66              5.8
4 4 2004-03-31            5           94   52     43     63              5.6
5 5 2004-04-30            4           96   54     37     59              5.6
6 6 2004-05-31            4           90   55     39     71              5.6
```

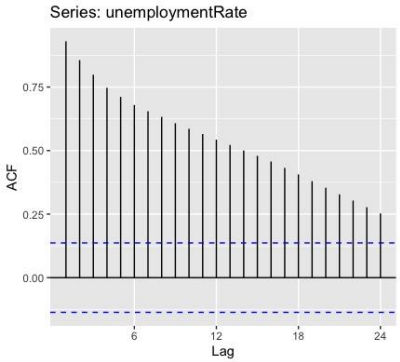


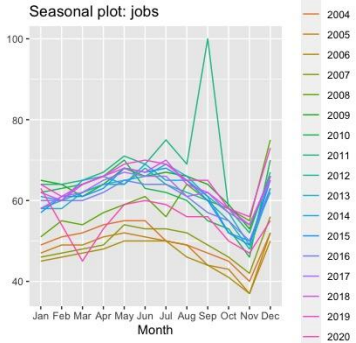


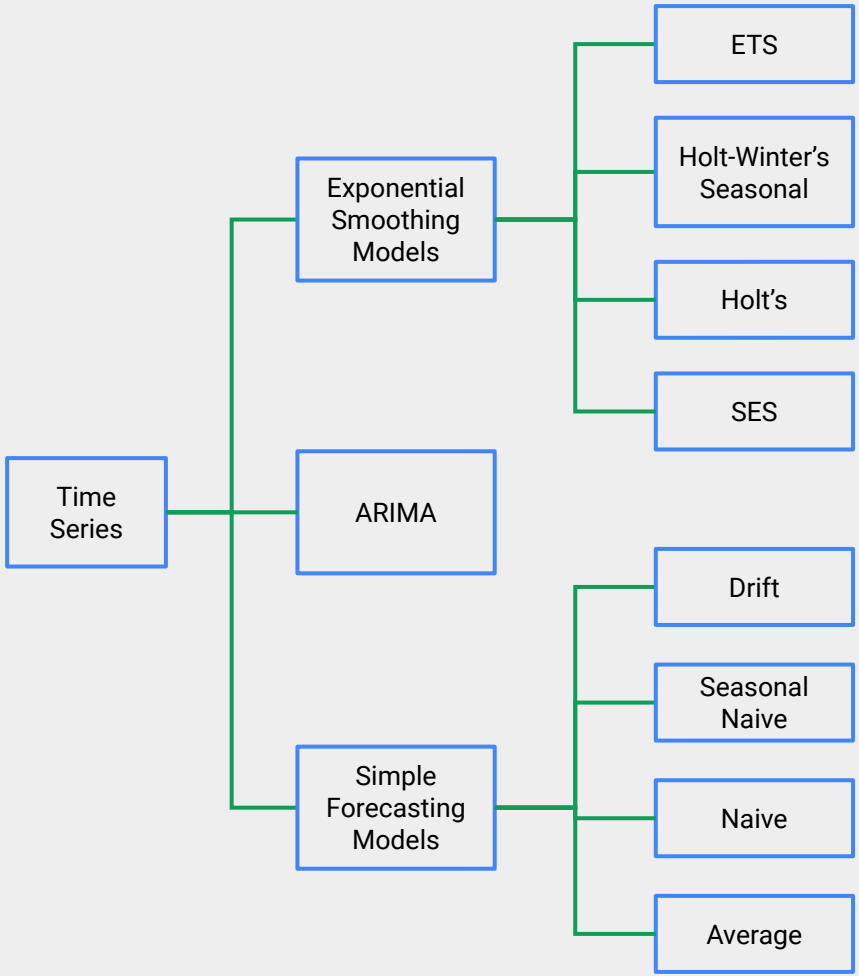
Analytical Technique 1:
Time Series









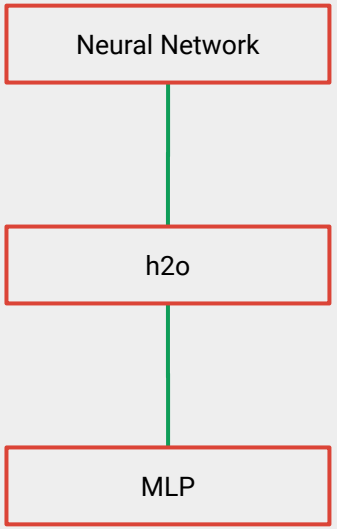


Analytical Technique 2: Neural Network

```
grid6 = h2o.grid(algorithm='deeplearning',
                 training_frame = train_h2o,
                 validation_frame=validation_h2o,
                 x=3:7,
                 y=8,
                 epochs=10,
                 stopping_metric='rmse',
                 stopping_tolerance=1e-2,
                 stopping_rounds=3,
                 hyper_params = hyper_parameters,
                 search_criteria = search_criteria)
 |=============================================================
best_model6 <- h2o.getModel(grid6@model_ids[[1]])

pred6 = h2o.predict(best_model6,test_h2o)
 |=============================================================
rmse(pred6[1],test_h2o$unemploymentRate)
1] 0.2165119
```

```
H2ORegressionModel: deeplearning
Model ID:  Grid_DeepLearning_train_sid_ad24_19_mode
Status of Neuron Layers: predicting unemploymentRat
0 training samples, mini-batch size 1
  layer units      type dropout       l1       l2 r
1     1     5     Input  0.00 %       NA       NA
2     2    25 Rectifier  0.00 % 0.000090 0.000050
3     3    25 Rectifier  0.00 % 0.000090 0.000050
4     4    25 Rectifier  0.00 % 0.000090 0.000050
5     5    25 Rectifier  0.00 % 0.000090 0.000050
6     6     1    Linear      NA 0.000090 0.000050
```



Analysis Results

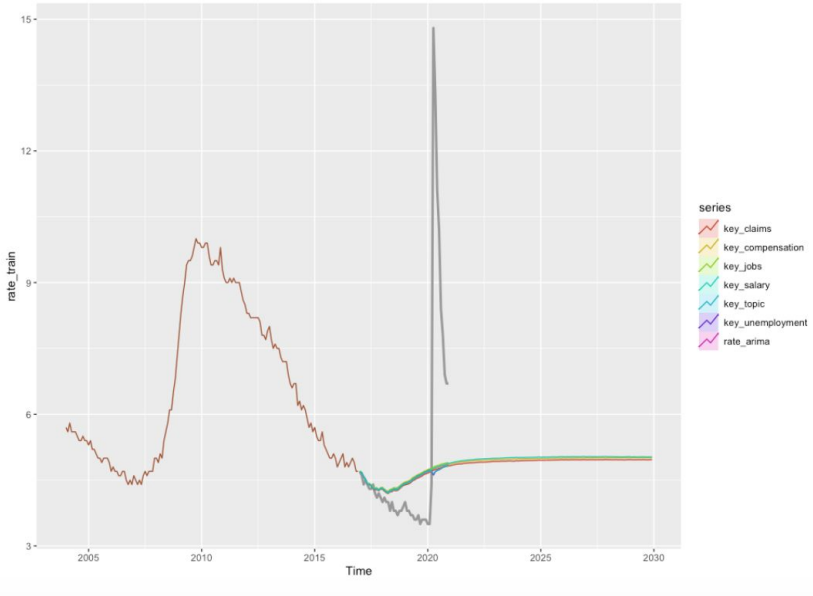
Best Keyword Predictor

- “Jobs”
- “Salary”

Best Model

- ARIMA
- Neural Network (overfitted)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
key_jobs	0.4616974	2.449776	1.300151	-2.821431	19.68844	1.395095	0.7773268	1.147862
key_salary	0.4914070	2.453036	1.280560	-2.116135	19.10642	1.374073	0.7770000	1.143281
key_compensation	0.4892639	2.457947	1.287626	-2.205320	19.25151	1.381656	0.7779112	1.145289
unemployment_rate	0.4941371	2.458282	1.284248	-2.089873	19.15705	1.378031	0.7778084	1.144709
key_claims	0.5208141	2.466255	1.271810	-1.492172	18.73181	1.364684	0.7782774	1.142783
key_topic	0.5034768	2.484308	1.296206	-2.020372	19.30567	1.390862	0.7790266	1.156537
key_unemployment	0.4958492	2.490102	1.304269	-2.237294	19.51376	1.399514	0.7788787	1.162321



Conclusion

Further Analyses

- Other Models
  - GARCH
  - VECM
  - LSTM
- Other predictors

Conclusion

- ARIMA with keywords was best model
- Some contradicting results with previous studies
- Our models will do better for nowcasting & with less volatility