# Byte Monsters

01000010 01111001 01110100 01100101 00100000 01001101 01101111
01101110 01110011 01110100 01100101 01110010 01110011 00001101

Isaac Thompson & Scott Tonidandel

SHAKER
virtualjobtryout®

DAVIDSON

# Machines Are Taking Over (or They Already Did).

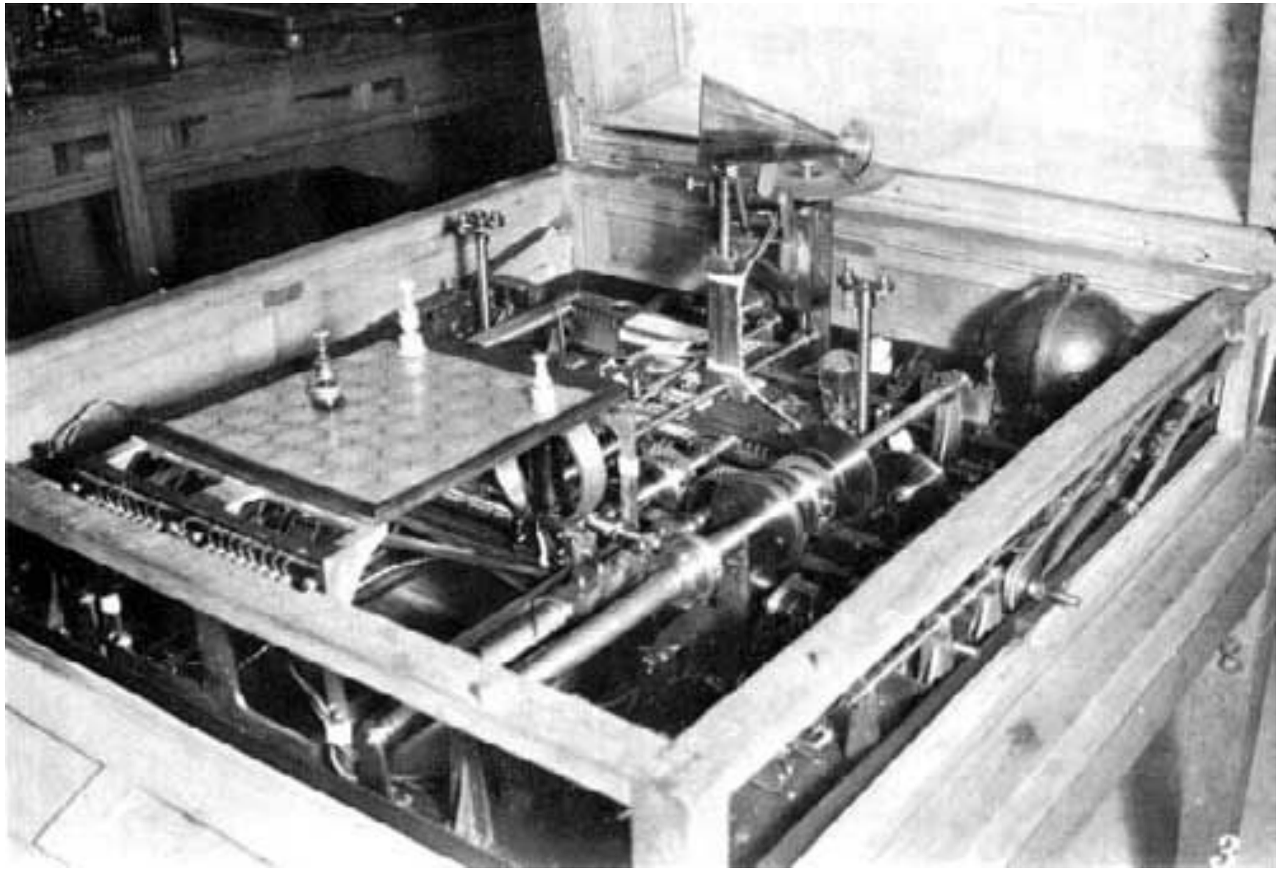
Automation and the future

Journey to leaderboard

Purpose: Enable I-Os

  Tricks of the trade

Takeaways



Second chess-automaton of Torres. The first made its debut in the Paris World Fair of 1914.

# Key Techniques

**Open source**

- Linux and Git
- R and Python
- XGBoost
  - Best in class prediction method (xgboost)

**Tricks of the trade**

- Hyperparameter tuning
- k-folds cross validation
- Feature engineering (variable creation)
- External data

# Open Source

Powering the data science and AI revolution.

- **Free**: scalable.
- **Secure**: multiple sources that are decentralized.
- **Transparent**: download source code.
- **Customizable**: modify the source code.
- **Collaborative:** constant development from multiple independent sources (on a global scale), in a state of constant improvement.
- **Innovative**: vastly more than proprietary software, offers a competitive advantage.

# Software Used

- Operating System
  - **Linux** (works on other ones too)
- Analytic software
  - **R**
- R Libraries:
  - **Dplyr** = great data handling program (get rid of excel)
  - **Xgboost** = e**X**treme **G**radient **Boost**ing
    - Open-source software library
    - Gradient boosting framework for C++, Java, Python, R, and Julia.
    - Works on Linux, Windows, and macOS.
    - Scalable, Portable and Distributed Gradient Boosting Library".
    - Can run on single machine (in parallel) or in distributed environments (such as Apache Hadoop, Apache Spark, and Apache Flink) .

# Data Magic

**Current data**

- Training data went from year #### to year #### .
- Goal to predict turnover of those folks 2009 to 2014.

**Feature engineering (creating data)**

- Folks 2009 to 2014 we have limited features (i.e. internal turnover rates for future by certain roles by country, job type, job function
- Imputation for missing data by median for continuous and unknown for categorical

**External data**

- Incorporated unemployment rate by country (of residence for 2009).

# Hyperparameter tuning

## Random parameters

```
hyper_params <- list(objective =
              "binary:logistic",
              eval_metric = "auc",
              max_depth = sample(6:10, 1),
              eta = runif(1, .01, .3),
              gamma = runif(1, 0.0, 0.2),
              subsample = runif(1, .6, .9),
              colsample_bytree = runif(1, .5, .8),
              min_child_weight = sample(1:40, 1))
```

## Iterate

- Ran it 250 times with random numbers
- Use xgboost cv with 8 folds to determine the optimal number of rounds to tune the xgboost
  - (to help with over fitting).

# Model Ensemble

- Our final model was an average of 3 Xgboost models

  1) Ran one model without external variables

  2) Ran one with very specific future turnover features

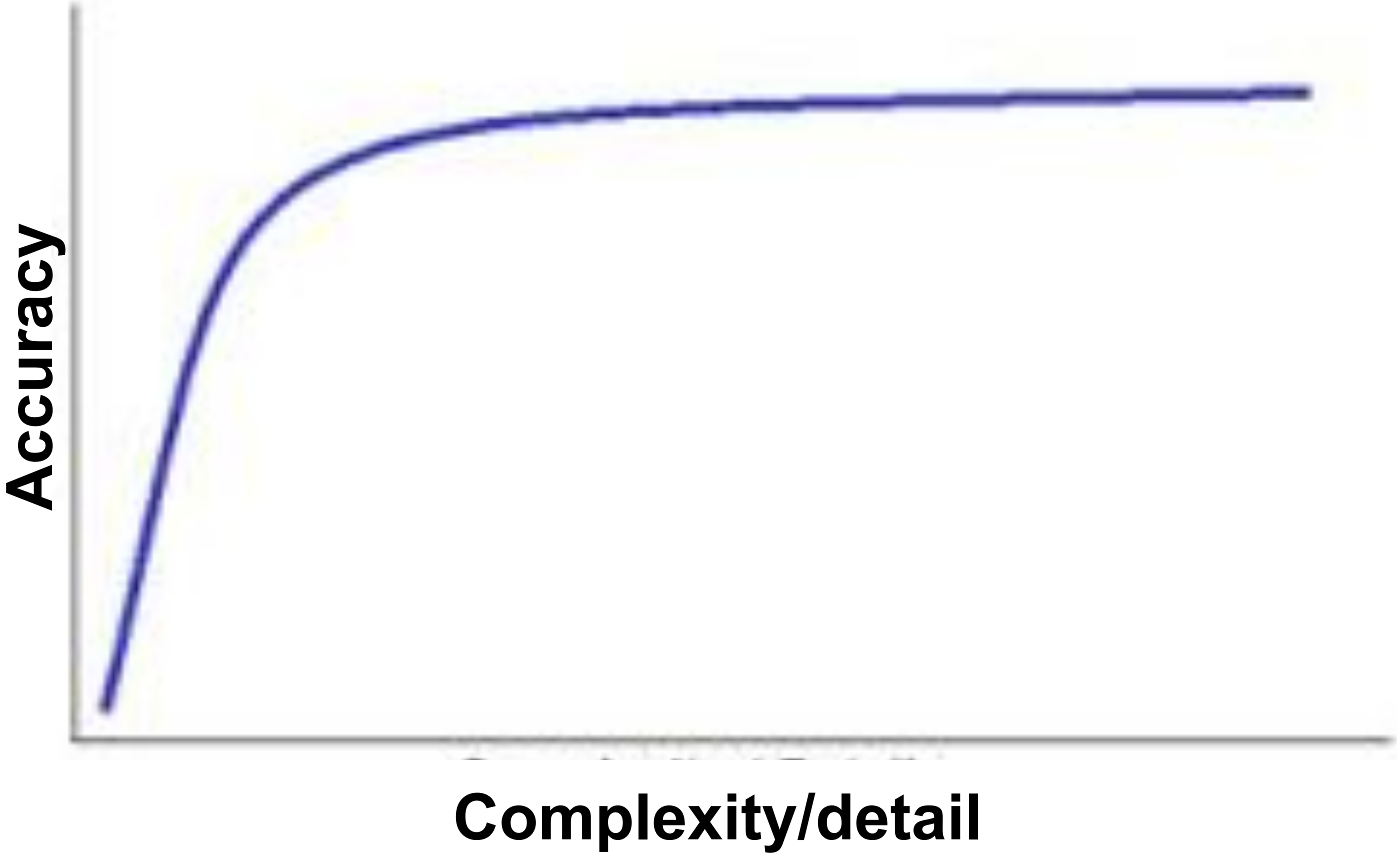
  3) Ran one with broad feature engineered future turnover variables

Averaged the predictions of the three (e.g. for one person predicted turnover would be .20, .21, .22, so ensemble prediction would be .21).

# Diminishing returns

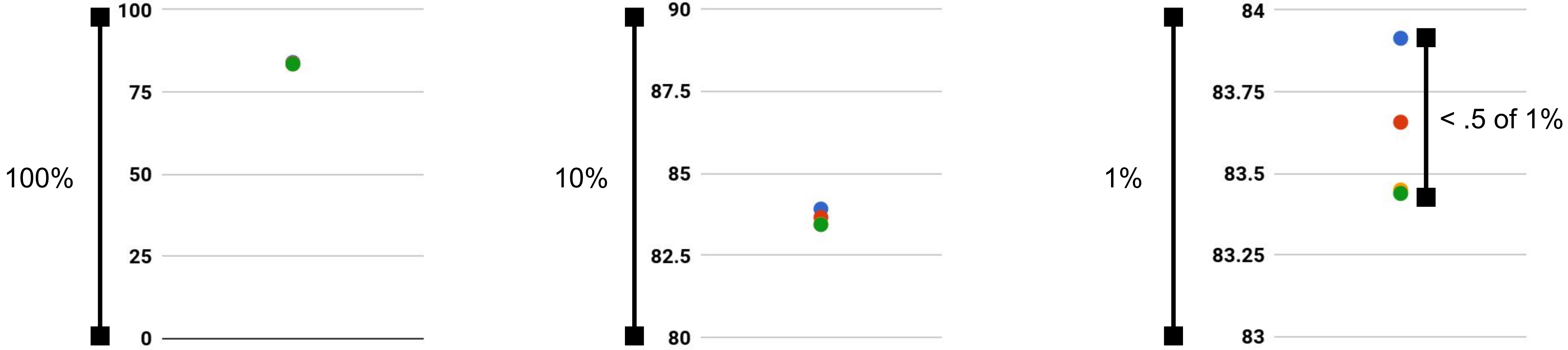
Red Hat’s Kaggle Competition

Any data competition



# Results: Competition Context

- Competitions: maximize accuracy quickly and then extremely incremental improvements line the leaderboard.
- IRL: too much model complexity or too small of incremental improvement may prohibit competition winning solutions from bringing IRL value.



**Takeaway**: depending on situation: take competion solutions with grain of salt. You can often have viable solutions that are 80% as good with 20% of complexity/effort

# Takeaway

Be lazy.

- Let the machines do the work.
- Test every possible solution, automatically.

Be a coder, better yet be a hacker.

- Open source is your best friend.
- Don’t ever be deterred.

Be creative.

- External data and feature engineering.

Be a learner.

- Rate of change = slope^10.

Be cutting edge.

- Latest greatest techniques.

# Thank you

Questions or comments

thompsonisaacb@gmail.com |sctonidandel@davidson.edu

code @ github.com/izk8