PyData Meetup

Sharing a Data Science side Project

Before I continue, after my sharing…

- I will share:
  - My slides & cleaned-up Github repo
  - My repo has my Jupyter notebooks, but no proper *requirements.txt*

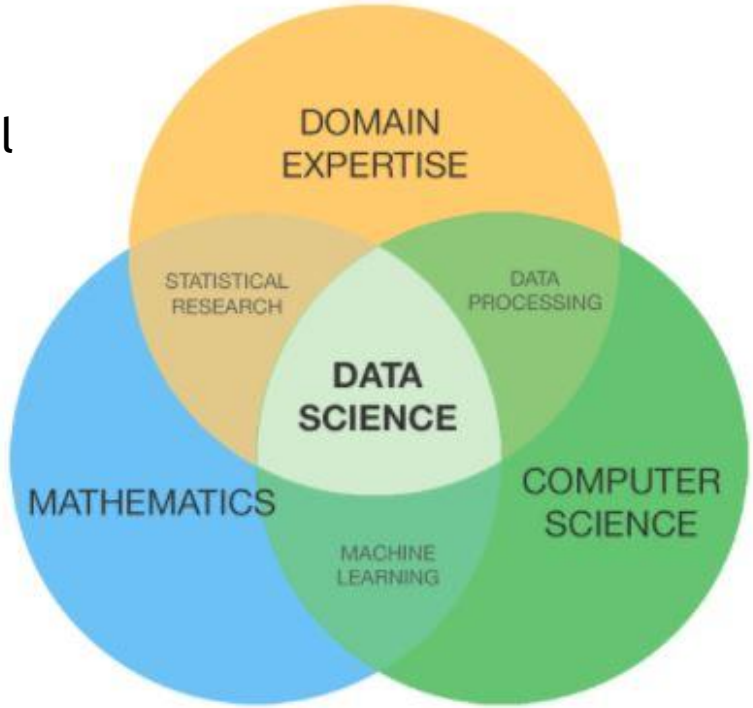
- I will not share:
  - My data set (because I have something else more useful)

Rough content for today’s sharing

- About Me + Today’s Focus
- Why my own DS side project?
- How I scope my project?
- My tech stack
- Getting data
- Cleaning data
- Training data
- Outputting data
- Code demo (lightly)
- Potential improvements
- What I learned
- If you want to start your own DS side project
- Parting Gift
- Q & A, if any

About me

- ***Name:*** Cliff Chew
- Digital Marketing Data Analyst @ Carousell
- Interested in (too) many pursuits!!!
- *Economics major with no formal technical training*
    - *Knew some statistics & ML*
    - *Self-learned some Python*
    - *No biz domain knowledge*
    - *<u>With so many gaps, how do I improve my DS skill sets?</u>*



Source: Palmer, Shelly. Data Science for the C-Suite. New York: Digital Living Press, 2015. Print.

Today's Focus

- Personal exploration & learnings from my own DS side project.
- Why you may want your own DS side project
- Some tips if you want to start one yourself

EMPHASIS!!

- *No new, advanced model*
- *No Deep Learning*
- *Nothing fancy*
- *Learn from the mistakes of others (mine today), or you risk having to learn from the mistakes of your own.*

Why I started my own DS side Project?

- **Flexibility**
  - Option to scope my own project
  - Less time pressure; No chasing KPIs (double-edged sword)
  - Easier & more flexibility than Kaggle

- **Improve my skill sets**
  - Hands-on ML learning + Practice Python coding
  - Exposure to data workflow & code maintenance work
  - Exploring different tech stacks!

How I scope my project?

Rule 1: I must be highly interested in the topic

- Much of mine leisure time will be replaced by this activity
- Can I keep up with other commitments in my life?

Rule 2: A classification or estimation problem on numbers

- As an Economics grad, I wanted to work on something familiar
- No text mining & image detection (future projects)

How I scope my project?

Rule 3: Data is available & plentiful but not overwhelming

- My laptop couldn't handle many Kaggle datasets.
- However, data must be available for ML to happen

Rule 4: Have some prediction output as a *"data product"*

- Don't let it be a single, static piece of analysis
- Build a workflow that allows me to create periodic predictions

My answers…

- **Topic: I love watching the NBA! Many start with sport analytics**
- **Data Availability: A quick Google showed that data is readily available**
  - I just need to get it, right? Easy??
- **What to analyse: Predict the probability of home team wins**
  - With data in my hands, I can do so many things: players clustering, impact of players after trades, etc…
- **Final output: Daily predictions of NBA games**
  - Rather than having a static analysis, maybe I can create an NBA daily prediction API! To an Economics major, this is totally mind-blowing!

Probability of home team wins of NBA Games

Quick Question

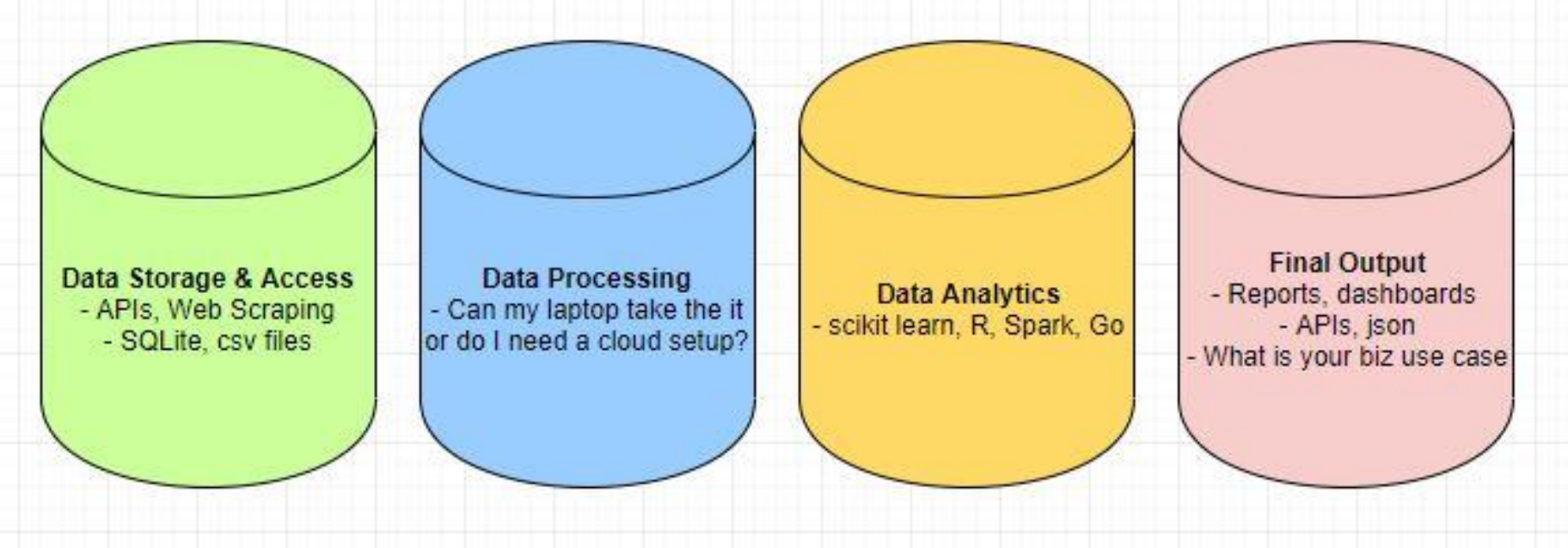
Any NBA fans here?

My (Eventual) Tech Stack

- **“Obtaining” data** → Selenium, BeautifulSoup, Tor
- **Data Storage** → CSV files (No database?)
- **Python** → Pandas, Scikit learn, Plotly (for viz)
- **Output** → Jupyter Notebooks, Github / Github Pages
- **Others** → Dropbox API, Google Sheets API

- **Bonus** → *I did explore the R as well if anyone is interested in it*

4 Pillars of Data Science



So, the data is available online, right?

I had to ask myself a lot of questions!

- Study the website's interface - How do I extract the data & what to extract?
  - Selenium + BeautifulSoup → Scraping
  - Tor → Web anonymity
- Is the data consistent across the different years?
- How do I store the scraped data?
  - Databases or just simple comma-separated files?
- How do I update my data from the site
- Designing the workflow for all this…

- ***Note:*** *Read up the legal & ethical concerns of scraping if you plan to do so!*

side story - My painful scraping experience

Cleaning Real World Data

- Inconsistent player IDs
- Teams / players name changes
- Rule changes in the NBA
- Advanced stats only exists recently
- Lockdown seasons have less regular games
- Playoffs had less games in the past

- ***I thought I had strong domain knowledge!***
- ***As the data was from 1 source, I thought it wouldn't be that dirty!***
- ***But interesting learnings happened!***

Preparing the data for model

- Wrangling raw data into training -ready format
- Calculating new features: Game counts, regular / playoff games
- Predicting only for home teams
- Calculating home / away / home opp / away opp stats
- Calculating cumulative stats for home and away teams
    - *E.g. To predict the probability of Team A winning its 10th game, which is a home game, I aggregate stats from all past 9 games played by Team A, against the aggregated stats of all games played by Team B up till the game with Team A. Confused?*

Why this approach?

- *E.g. To predict the probability of Team A winning its 10th game, which is a home game, I aggregate stats from all past 9 games played by Team A, against the aggregated stats of all games played by Team B up till the game with Team A. Confused?*
- Focusing on home wins allow me to focus on games, not teams.
- The cumulative stats allow me to use information from all past games played in that season.

Data for Model Training

- Data: 33,801 regular season NBA games from 1986-2016
- 1986-2015 training; 2016 test
- Predictions based on *(1) home team stats*, *(2) home team opponent stats*, *(3) away team stats* & *(4) away team opponent stats*
- Variables of current model include…
  - **Basic stats:** points, assists, offensive rebounds, defensive rebounds, turnovers, steals, blocks…
  - **Advanced stats:** efficient field goal %, free throw %, field goal %, 3-point field goal %...

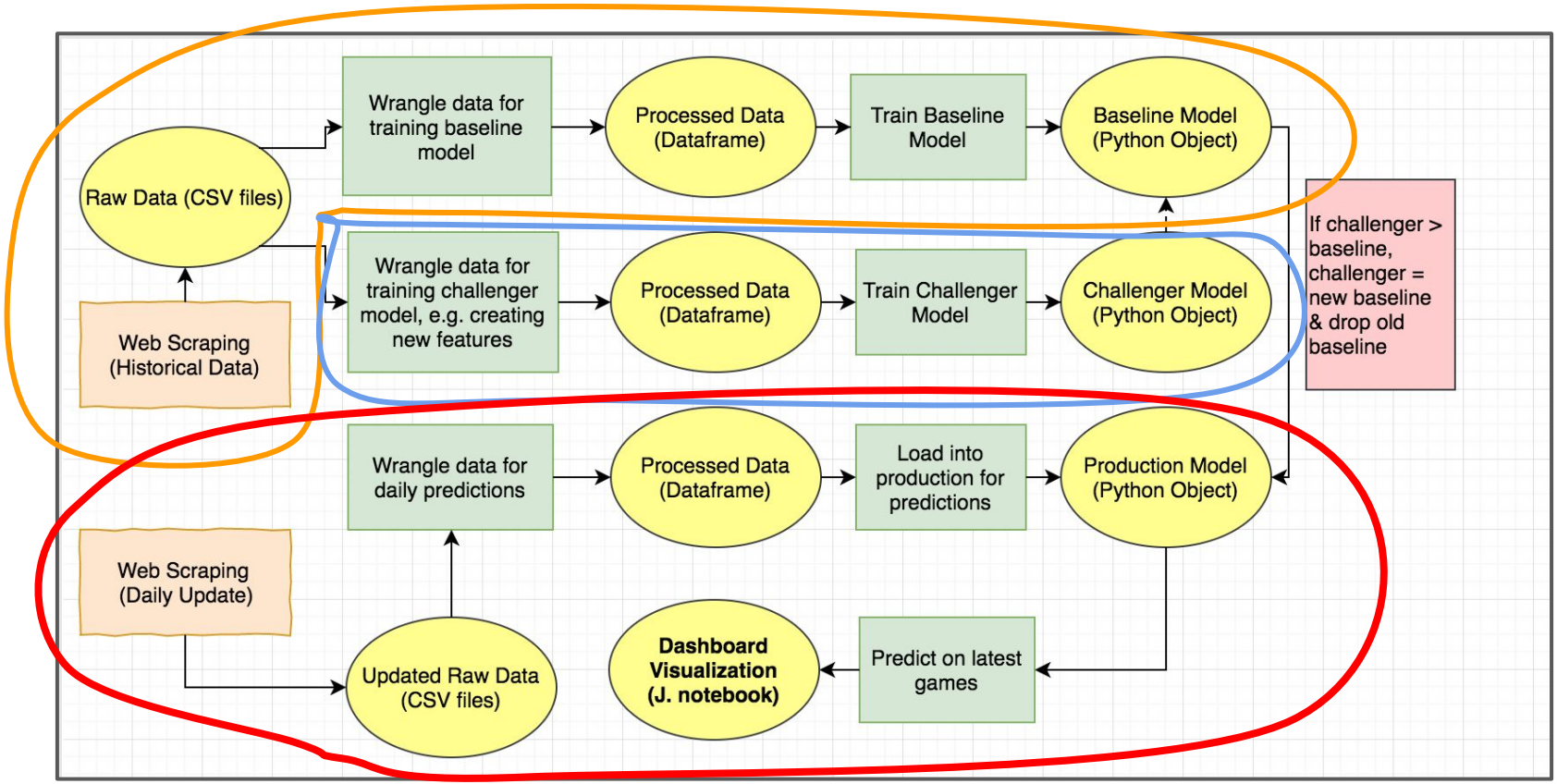
Model Training

- ***Models:*** Logistic regression, SVM, random forest, Adaboost, XGboost.
  - A few months of on-off training and testing…
  - Pushed results to Google Sheets through Google API
- ***Final results*** → Logistic regression gave the highest accuracy (<60%)
- *Saved the prediction model as a Python Object*
- I can always work on a challenger model thereafter
- Set up daily predictions workflow!

Workflow for Daily Predictions

- ***So my prediction model is out, but now what?***
- Modified code to perform:
  - Daily games stats updates (scraping script)
  - Daily predictions (model prediction script)
- Create Plotly dashboard to evaluate my production model's performance
- Other changes made:
  - Simplified inputs of production model for my prediction workflow!
  - Extract schedule of NBA current season

Current workflow



Code Demo ⟶ *I hope this will not be too **underwhelming!***

WiP(s) Improvements

- Piecing together *player data* to account for injuries & player trades
- *Player clustering* - Identifying players types so as to form team types
- *Ensemble training*
- Use more recent, extensive data: E.g. Play-by-play data, fast-break points…
- Switch *from a classification problem* to a *score estimation problem*.
    - Estimate scores of home and away teams based on the data, and calculate wins & losses from there
- Automating predictions - Airflow?
- Make components modular! Switch models / inputs easily!

What I learned?

- **Scrape only as a last resort.** If possible, use APIs instead!
- Sensitivity to real-world data; **Real-world data is really dirty**
- **Automate** as much stuff as you can; **Focus on fast iterations**
- *A working, “deployed” model v.s. perfecting a never “deployed” model*
- **Don’t be fixated on 1 tool.** Use the right tool for the right problem
- **Structured guidance is still necessary**. *virtualenv*, *git*, *quality code structure*

Starting your own DS project…

- ***Motivations***: What are your motivations for this? Good advice for anything to be honest
- ***Domain knowledge***: Passion in the topic keeps things lively; Focus on technical learnings
- ***Data availability.*** No data, no ML. No clean data, no point doing ML
    - With data at hand, I could do other interesting stuff too: http://cliffchew84.github.io/side_story_playoffs.html
- ***Output***. What outcome / output you want to achieve in the end?

For those interested in sports analytics...

Throne AI is the Kaggle of Sports Predictions

By George McIntire, ODSC | 01/18/2018

My big obsession of 2018 so far is sports prediction platform Throne AI. There's no better way to describe than Kaggle for sports. The platform provides users with data with which they use to build models to predict the outcome of sports matches. Each league on Throne AI counts as its own competition with its own ranking of users. It currently has the following league available: NFL, NBA, NHL, English Premier League, Serie A, La Liga, and the English Championship, with more to come.

https://opendatascience.com/blog/throne-ai-is-the-kaggle-of-sports-predictions

The END - Questions?