NBA MATCHUP MACHINE

GITHUB REPO

DATS-6450 - CLOUD COMPUTING

TEAM 7: ALEXIS KALDANY,
SUHAS BURAVALLA, NATE EHAT

TABLE OF CONTENTS

01

OVERVIEW / BACKGROUND

02

IDEOLOGY / METHODOLOGY

03

MODELING PIPELINE

04

AWS ARCHITECTURE

05

CONCLUSIONS

06

APPENDIX

OVERVIEW / BACKGROUND

MOTIVATIONAL QUOTES

- “It's not about any one person. You've got to get over yourself and realize that it takes a group to get things done” – Greg Popovich, HC, San Antonio Spurs
- “Some people want it to happen, some wish it would happen, others make it happen.” – Michael Jordan, #23, Chicago Bulls
- “You always have to be on edge. You always have to take every practice, every game, like it is your last.” – Kobe Bryant, #8/#24, Los Angeles Lakers

NBA OVERVIEW

- Founded in 1946, the NBA began operations with 11 original franchises.
- Following multiple league expansions and a handful of franchise relocations, current league structure consists of 30 teams, located across the US and Canada:
    - 2 Conferences: Eastern / Western
    - 6 Divisions: Atlantic / Central / Southeast / Northwest / Pacific / Southwest
    - 82 regular season games (41 Home / 41 Away)
    - 10 teams with best record in each conference advance to league playoffs
    - 7-game playoff series, with first team to win 4 games advancing
    - Eastern / Western Conference Champions face off for NBA title

IDEOLOGY / METHODOLOGY

ANALYSIS BACKGROUND

- Given the inherently unpredictable nature of professional sports, generating accurate predictions for specific matchup outcomes has proven difficult for professional industry-leading statisticians and even NBA general managers.

- Research team hopes to contribute meaningful insights to the NBA data science community by providing access to predictions free of charge

DATASET

- By casting a wide net across various indicators or measurements of NBA team performance over time, the research team hopes to aggregate, synthesize, and iteratively re-weight historical team metrics to offer daily matchup predictions.

- 30 TEAMS
- 82 GAMES / SEASON

- 6 SEASONS [2014-2021]
- 14,700+ MATCHUPS

MODELING PIPELINE

MODELING PIPELINE

1) **Scraping / Wrangling – historical team data; advanced team metrics**

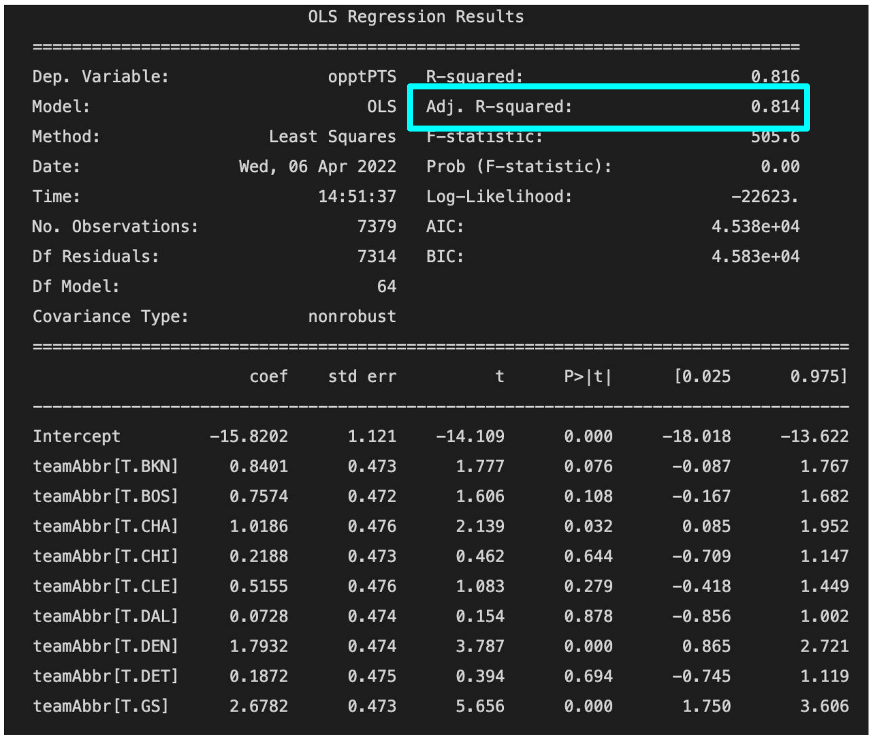
2) **Cleaning / Pre-Processing – aggregate and synthesize historical game records**

3) **Feature Engineering – incorporate detailed historical matchup box scores**

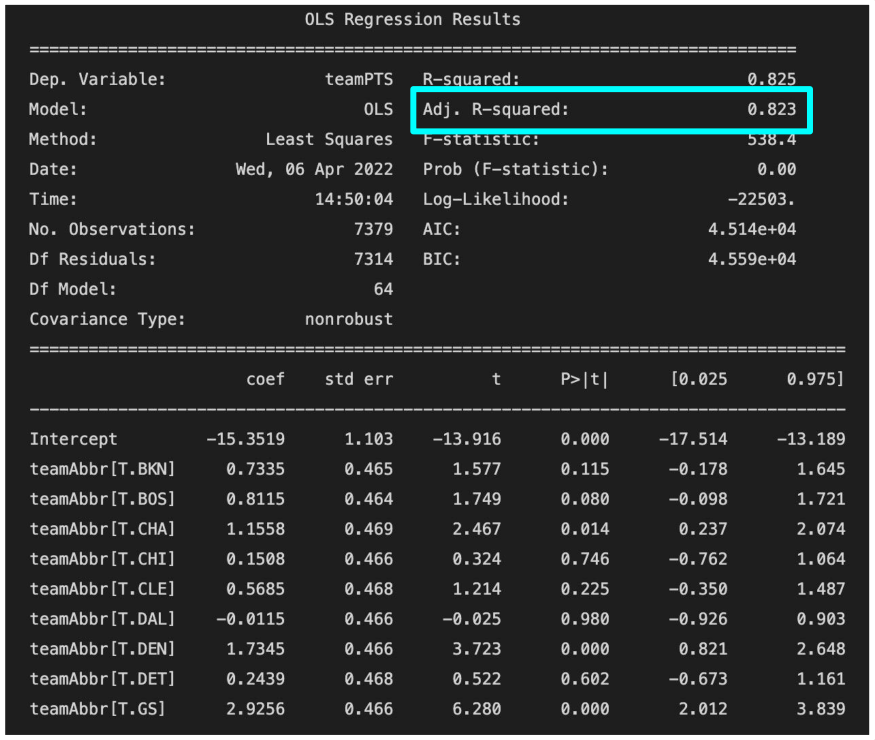
4) **Predictions – Regression model to generate predicted matchup outcomes**

5) **Deployment / Integration – implement model across AWS cloud infrastructure**

MODEL DETAIL

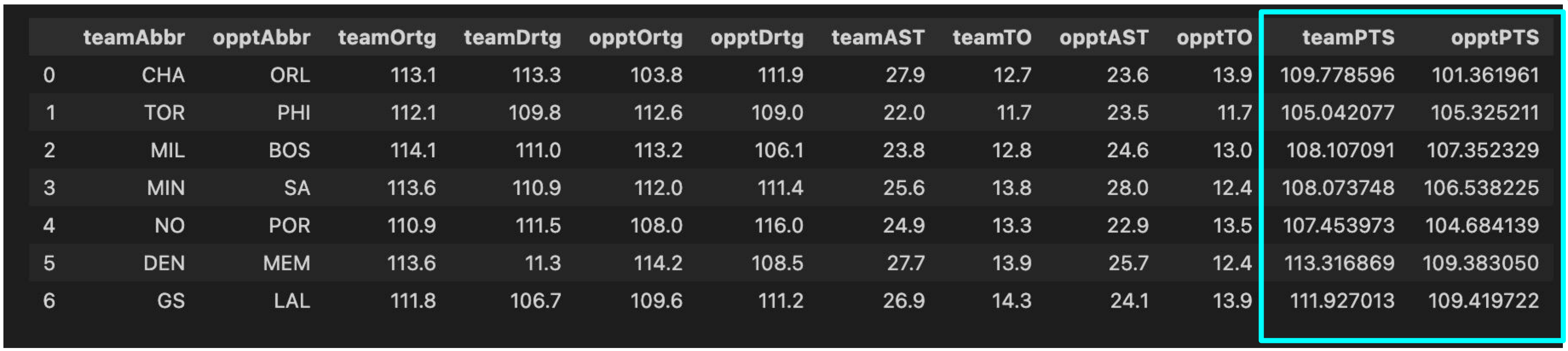


```
                            OLS Regression Results
==============================================================================
Dep. Variable:                opptPTS   R-squared:                       0.816
Model:                            OLS   Adj. R-squared:                  0.814
Method:                 Least Squares   F-statistic:                     505.6
Date:                Wed, 06 Apr 2022   Prob (F-statistic):               0.00
Time:                        14:51:37   Log-Likelihood:                -22623.
No. Observations:                7379   AIC:                         4.538e+04
Df Residuals:                    7314   BIC:                         4.583e+04
Df Model:                          64
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept          -15.8202      1.121    -14.109      0.000     -18.018     -13.622
teamAbbr[T.BKN]      0.8401      0.473      1.777      0.076      -0.087       1.767
teamAbbr[T.BOS]      0.7574      0.472      1.606      0.108      -0.167       1.682
teamAbbr[T.CHA]      1.0186      0.476      2.139      0.032       0.085       1.952
teamAbbr[T.CHI]      0.2188      0.473      0.462      0.644      -0.709       1.147
teamAbbr[T.CLE]      0.5155      0.476      1.083      0.279      -0.418       1.449
teamAbbr[T.DAL]      0.0728      0.474      0.154      0.878      -0.856       1.002
teamAbbr[T.DEN]      1.7932      0.474      3.787      0.000       0.865       2.721
teamAbbr[T.DET]      0.1872      0.475      0.394      0.694      -0.745       1.119
teamAbbr[T.GS]       2.6782      0.473      5.656      0.000       1.750       3.606
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                teamPTS   R-squared:                       0.825
Model:                            OLS   Adj. R-squared:                  0.823
Method:                 Least Squares   F-statistic:                     538.4
Date:                Wed, 06 Apr 2022   Prob (F-statistic):               0.00
Time:                        14:50:04   Log-Likelihood:                -22503.
No. Observations:                7379   AIC:                         4.514e+04
Df Residuals:                    7314   BIC:                         4.559e+04
Df Model:                          64
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept          -15.3519      1.103    -13.916      0.000     -17.514     -13.189
teamAbbr[T.BKN]      0.7335      0.465      1.577      0.115      -0.178       1.645
teamAbbr[T.BOS]      0.8115      0.464      1.749      0.080      -0.098       1.721
teamAbbr[T.CHA]      1.1558      0.469      2.467      0.014       0.237       2.074
teamAbbr[T.CHI]      0.1508      0.466      0.324      0.746      -0.762       1.064
teamAbbr[T.CLE]      0.5685      0.468      1.214      0.225      -0.350       1.487
teamAbbr[T.DAL]     -0.0115      0.466     -0.025      0.980      -0.926       0.903
teamAbbr[T.DEN]      1.7345      0.466      3.723      0.000       0.821       2.648
teamAbbr[T.DET]      0.2439      0.468      0.522      0.602      -0.673       1.161
teamAbbr[T.GS]       2.9256      0.466      6.280      0.000       2.012       3.839
```

MODEL PREDICTIONS



	teamAbbr	opptAbbr	teamOrtg	teamDrtg	opptOrtg	opptDrtg	teamAST	teamTO	opptAST	opptTO	teamPTS	opptPTS
0	CHA	ORL	113.1	113.3	103.8	111.9	27.9	12.7	23.6	13.9	109.778596	101.361961
1	TOR	PHI	112.1	109.8	112.6	109.0	22.0	11.7	23.5	11.7	105.042077	105.325211
2	MIL	BOS	114.1	111.0	113.2	106.1	23.8	12.8	24.6	13.0	108.107091	107.352329
3	MIN	SA	113.6	110.9	112.0	111.4	25.6	13.8	28.0	12.4	108.073748	106.538225
4	NO	POR	110.9	111.5	108.0	116.0	24.9	13.3	22.9	13.5	107.453973	104.684139
5	DEN	MEM	113.6	11.3	114.2	108.5	27.7	13.9	25.7	12.4	113.316869	109.383050
6	GS	LAL	111.8	106.7	109.6	111.2	26.9	14.3	24.1	13.9	111.927013	109.419722

AWS ARCHITECTURE

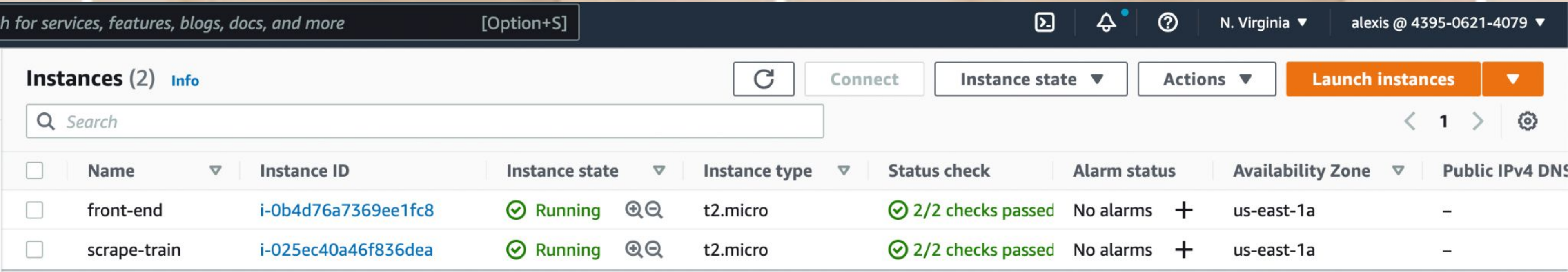
AWS ARCHITECTURE

- EC2-1 scrapes data and runs model. Reads and writes to S3 bucket
- EC2-2 hosts our Plotly front-end application. Reads from S3 bucket.
- S3: Use a single S3 bucket. Contains raw data, model outputs, and transformed model outputs which are displayed on front-end.

Version 2.0

- Using Elastic Load Balancing to enable scaling of the front-end
- Using a workflow engine like Airflow to build a pipeline to automate scraping + modeling tasks, this would require setting up a Kubernetes cluster and Dockerizing the application.
- Replacing Plotly-Dash front-end with a Django web-framework
- Django would map to a PostGres managed RDS on AWS, which would contain all tabular data. S3 would contain graphics/videos embedded on front-end.

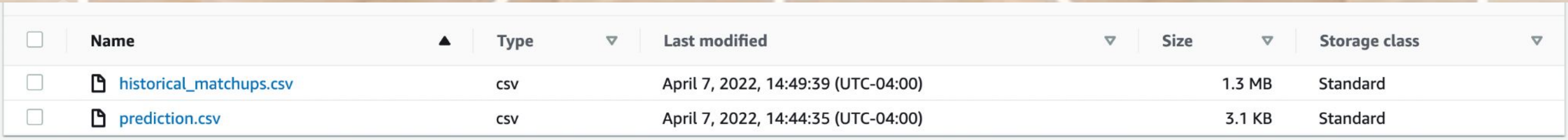
EC2 Console



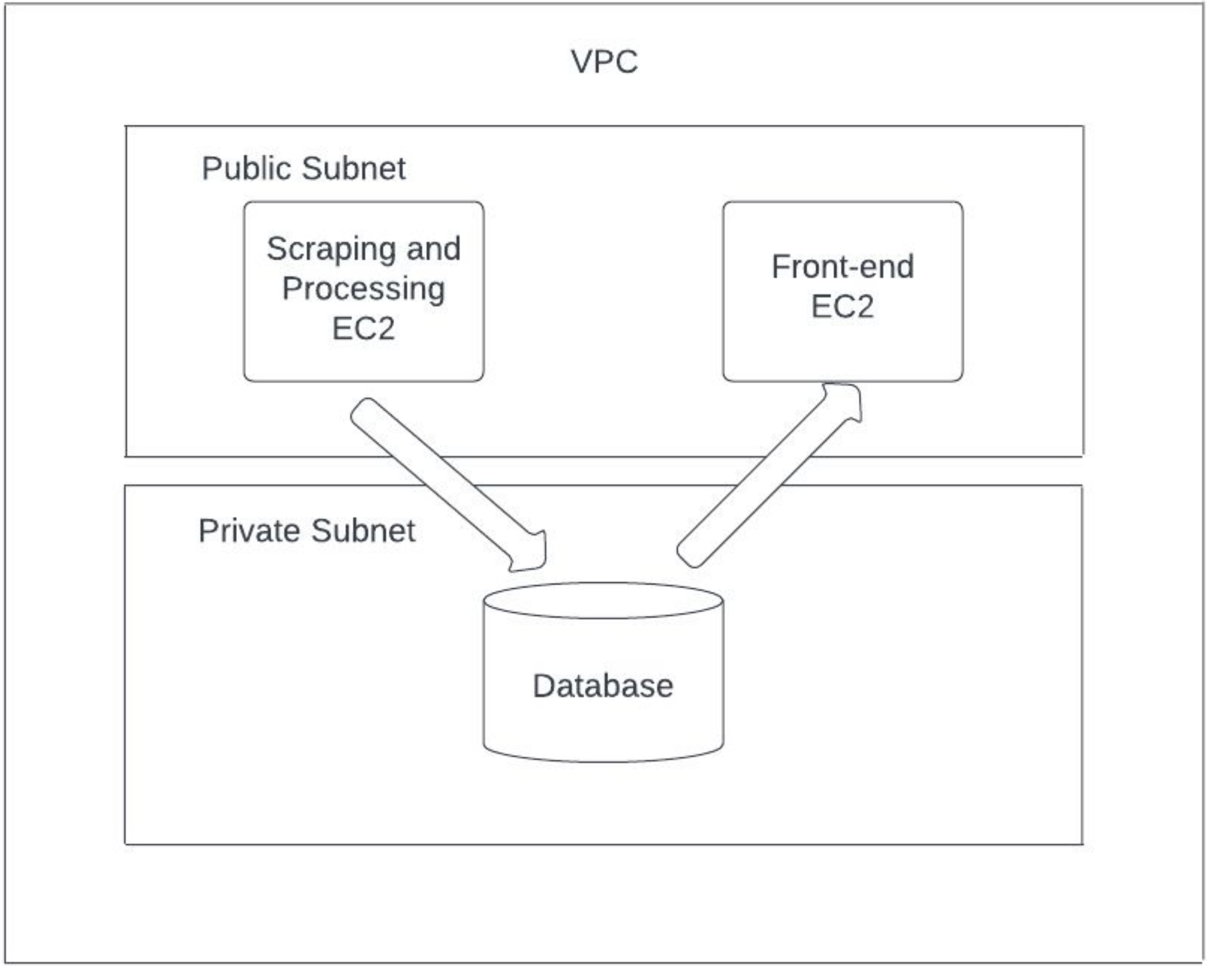
Instances (2) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	front-end	i-0b4d76a7369ee1fc8	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1a	–
□	scrape-train	i-025ec40a46f836dea	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1a	–

S3 Project Bucket



□	Name ▲	Type	Last modified	Size	Storage class
□	historical_matchups.csv	csv	April 7, 2022, 14:49:39 (UTC-04:00)	1.3 MB	Standard
□	prediction.csv	csv	April 7, 2022, 14:44:35 (UTC-04:00)	3.1 KB	Standard


VPC
Public Subnet
Scraping and Processing EC2
Front-end EC2
Private Subnet
Database

CONCLUSIONS

CONCLUSIONS / TAKEAWAYS

- **DATA MINING:**
  - Wide availability of data provides a diverse menu of statistical features
  - Significant focus on controlling / reducing multi-collinearity of data points
- **MODEL PIPELINE:**
  - Regression models are better suited for predicting matchup wins
- **AWS CLOUD:**
  - Current deployment sufficient as demonstration
  - Scaling requires integration of additional services / workflow management tools

FUTURE PRODUCT ROADMAP

- **DATA MINING:**
  - Dynamic lineup adjustments to reflect real-time lineups / injury reports
  - Implement certain ‘intangible’ or auxiliary statistics (coach, travel time, rest)
- **MODEL PIPELINE:**
  - Integrate player-level database to improve team-level predictions
  - Refine proprietary benchmark metric to further optimize model performance
    - SUHAS’ SECRET SAUCE ™℠®©
- **AWS CLOUD:**
  - Link / connect storage buckets to preserve incremental player-level data
  - Launch ‘product-ready’ dashboard interface + mobile app

APPENDIX

SOURCES / CITATIONS

- Basketball Reference – https://www.basketball-reference.com/
- Team Rankings – https://www.teamrankings.com/nba/stats/
- NBA.com – https://www.nba.com/stats/teams/traditional/
- NBA.com API – https://pypi.org/project/nba-api/
- Basketball Monster – https://basketballmonster.com/playernews.aspx
- Bball Index – https://www.bball-index.com/player-impact-plus-minus/
- Bball Index – https://www.bball-index.com/lebron-introduction/
- FiveThirtyEight – https://projects.fivethirtyeight.com/nba-player-ratings/