Rinnovation

Innovate Your Home Renovation

Team

Aditya

Andrew

Kaustubh

Pallavi

“Measure twice, cut once.”

-Carpenters

Personas

Home Owners

Home Buyers

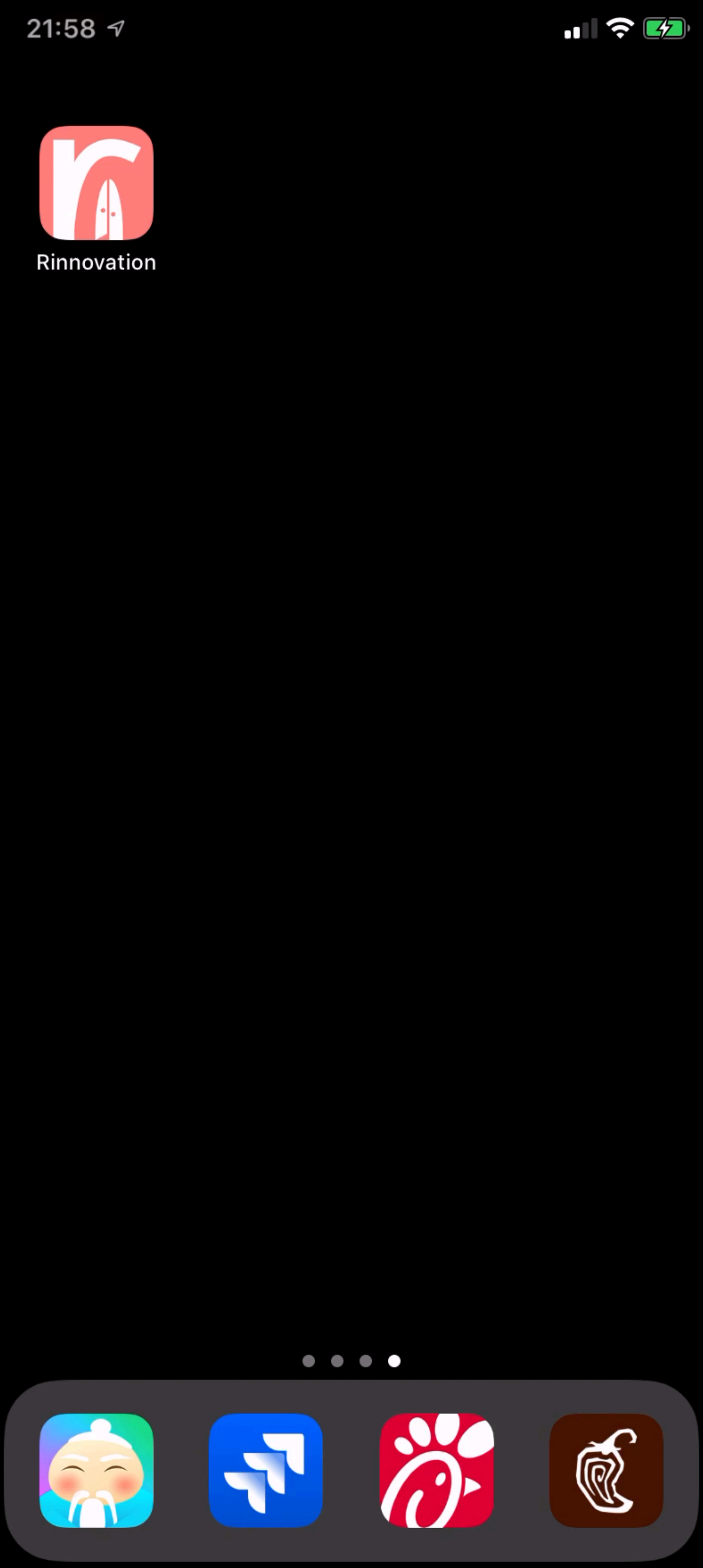
iBuyer Executives

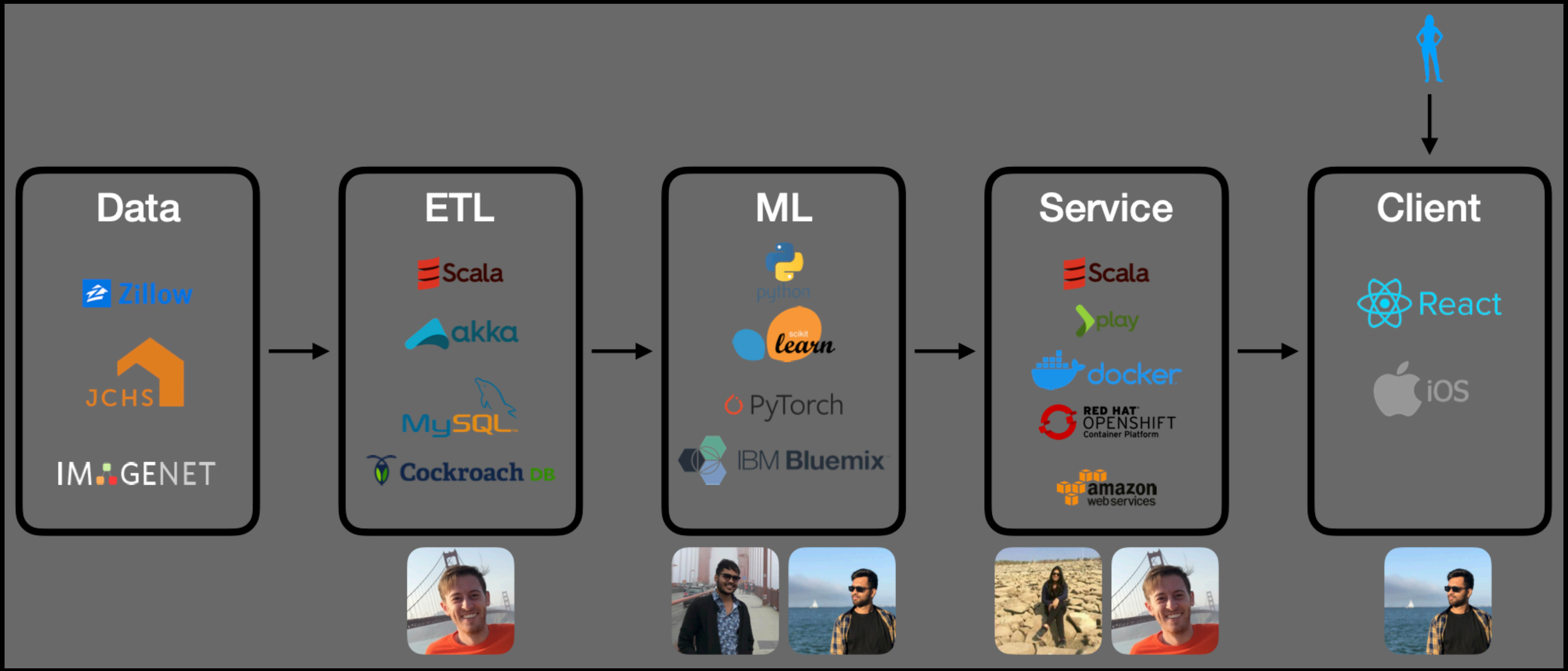
Goals

Enable our 3 personas to:

- View historical renovation outcomes
- Predict ROI for specific types of renovations
- See trends tailored down to individual cities

Demo


21:58
Rinnovation



Initial Architecture

Data

Data

Zillow

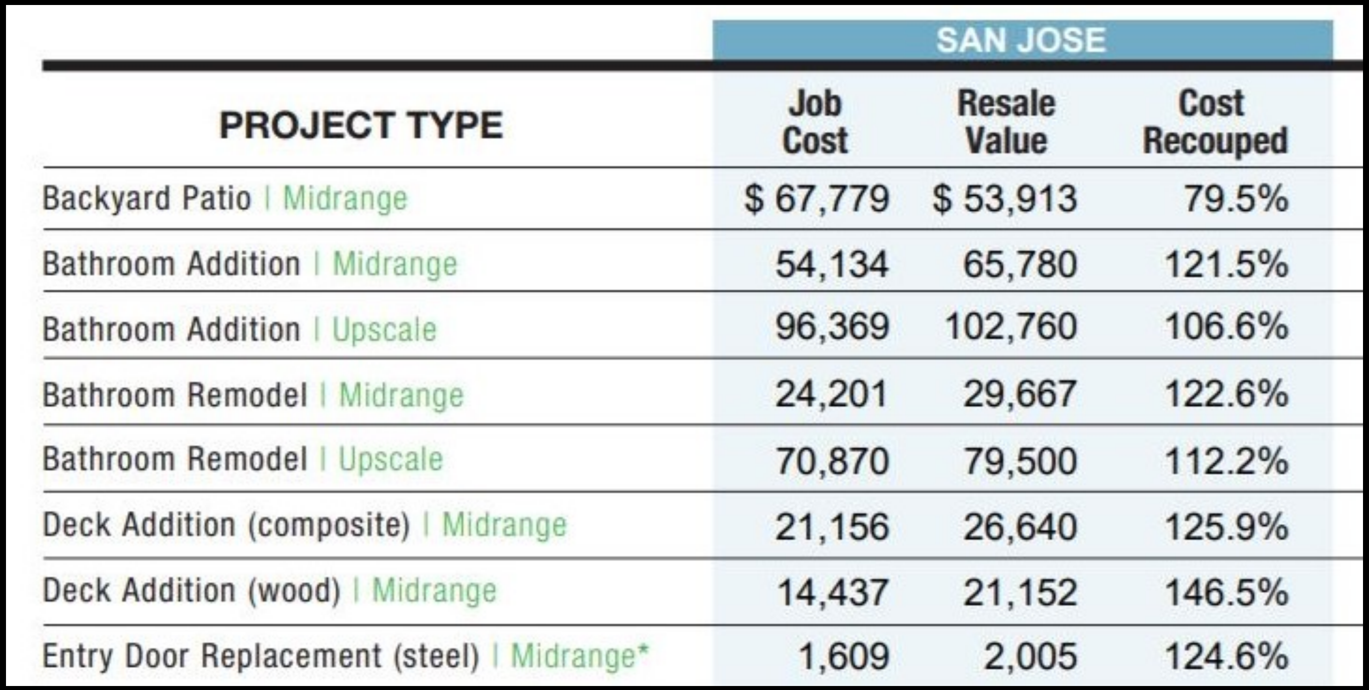
JCHS

IMAGENET

- Zillow: not enough historical data
- JCHS: focused mostly on demographics
- ImageNet: not core to our mission

Eureka!

We finally discovered a data set (Cost vs. Value) with city-specific renovation outcomes for cities across the US.



	SAN JOSE		
PROJECT TYPE	Job Cost	Resale Value	Cost Recouped
Backyard Patio I Midrange	$ 67,779	$ 53,913	79.5%
Bathroom Addition I Midrange	54,134	65,780	121.5%
Bathroom Addition I Upscale	96,369	102,760	106.6%
Bathroom Remodel I Midrange	24,201	29,667	122.6%
Bathroom Remodel I Upscale	70,870	79,500	112.2%
Deck Addition (composite) I Midrange	21,156	26,640	125.9%
Deck Addition (wood) I Midrange	14,437	21,152	146.5%
Entry Door Replacement (steel) I Midrange*	1,609	2,005	124.6%

Issues

- Only published as PDFs, not plain text
- Slight format variations between each year

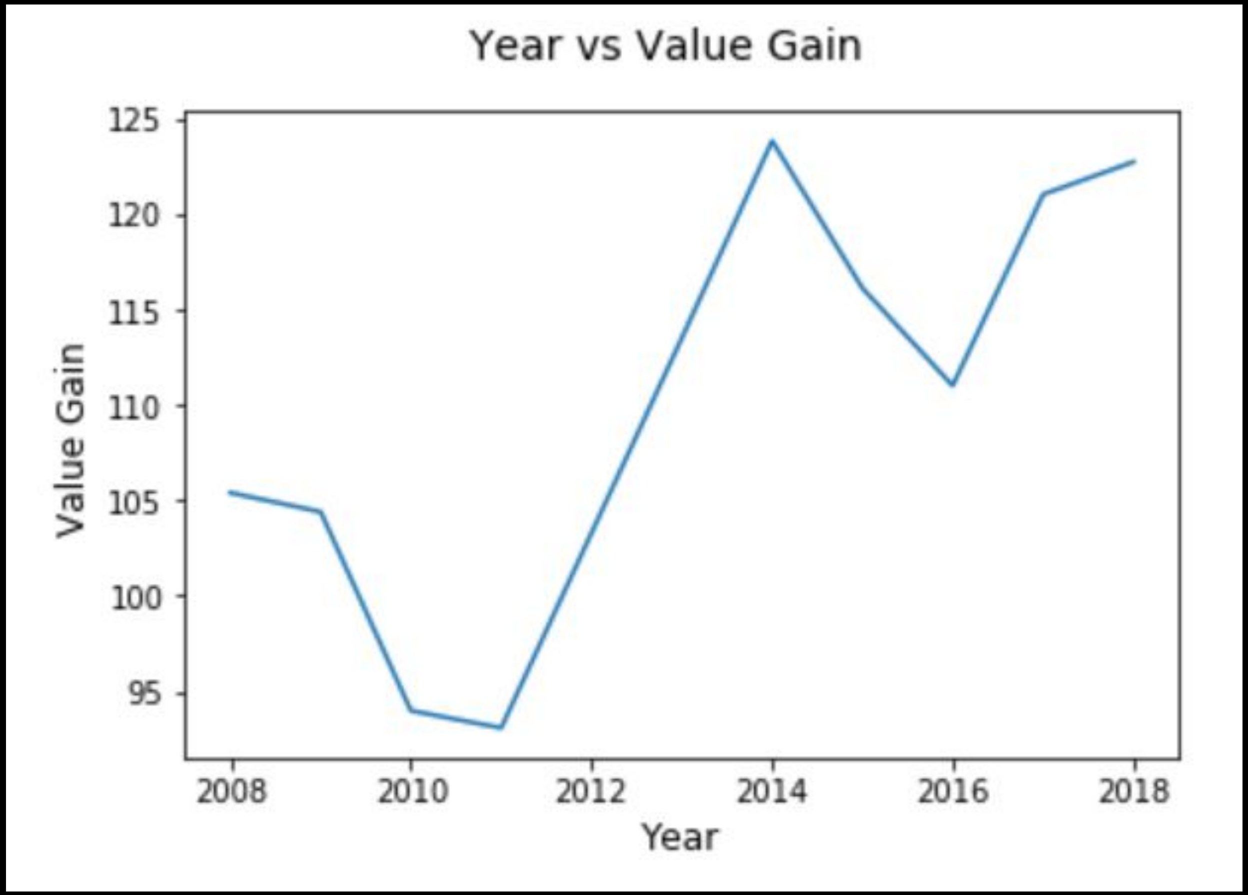
Solutions

- tabula-java: Extracts CSVs from tables in PDFs
- scala-csv: Helps massage imperfect CSVs

Data Processing

- The data we had was not stationary. We used **Kwiatkowski-Phillips-Schmidt-Shin (KPSS)** and **Augmented Dickey–Fuller (ADF)** test to check for Stationarity of data. We got to know our data wasn't strictly stationary, but trend stationary.

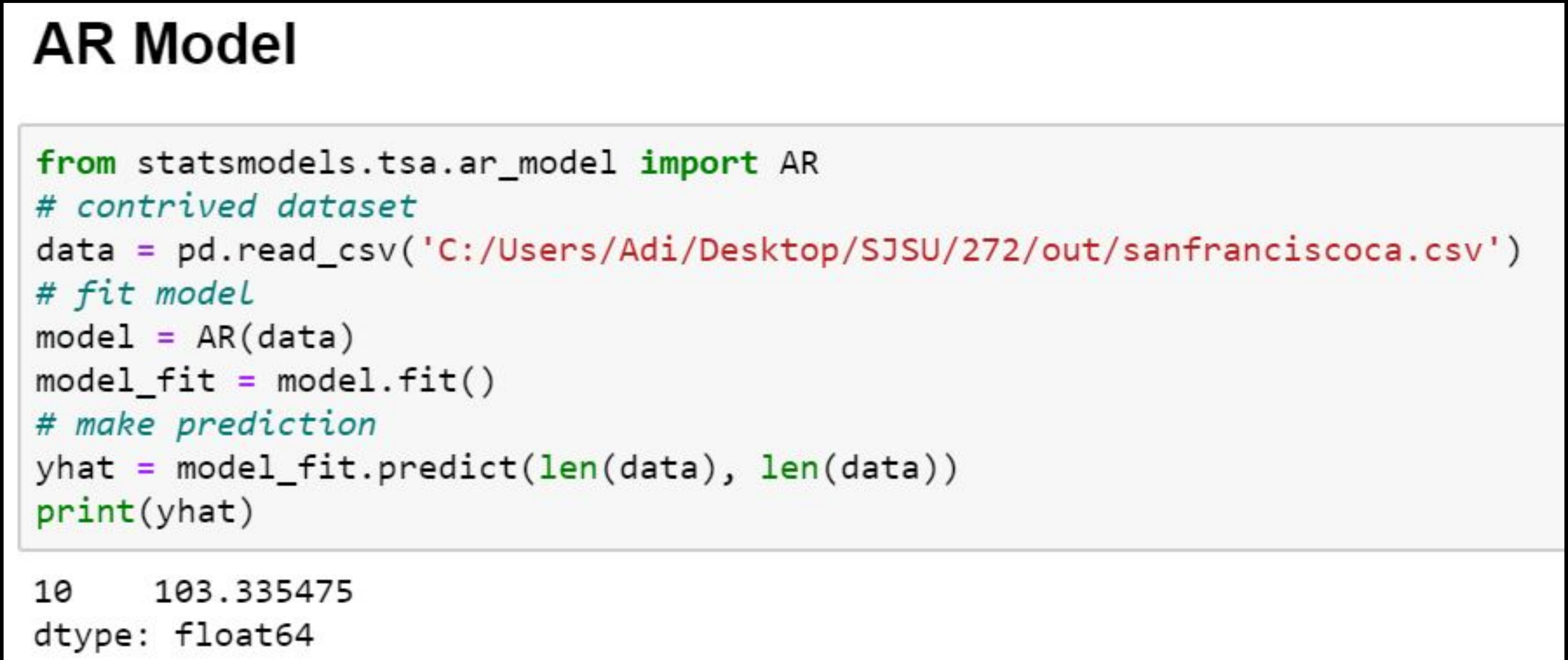
- We did "Differencing" and then "Log Transformation" to make the data strict .



Data Modeling

- We tried 5 models for Time Series prediction: MA, ARMA, ARIMA, SARIMA, but ultimately chose Autoregression (AR)

- We tested these by running in values for years 2008 to 2017 and checking how close the values were to actual values of 2018



AR Model

```
from statsmodels.tsa.ar_model import AR
# contrived dataset
data = pd.read_csv('C:/Users/Adi/Desktop/SJSU/272/out/sanfranciscoca.csv')
# fit model
model = AR(data)
model_fit = model.fit()
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

```
10    103.335475
dtype: float64
```

ETL

ETL
Scala
akka
MySQL
Cockroach DB

- A collection of discrete Scala applications
- Extracts PDFs for each year 2008-2019
- 150 cities processed
- Google Maps Geocoding API
- Raw PDFs transformed into:
    - CSVs for consumption by ML
    - SQL for loading into Postgres

Service

- Written in Scala
- Built atop the Play framework
- Built & run with Docker
- Images pushed to Docker Hub: https://hub.docker.com/repository/docker/rinnovation/rinnovation-service
- Deployed to Amazon ECS

Client

Client

React

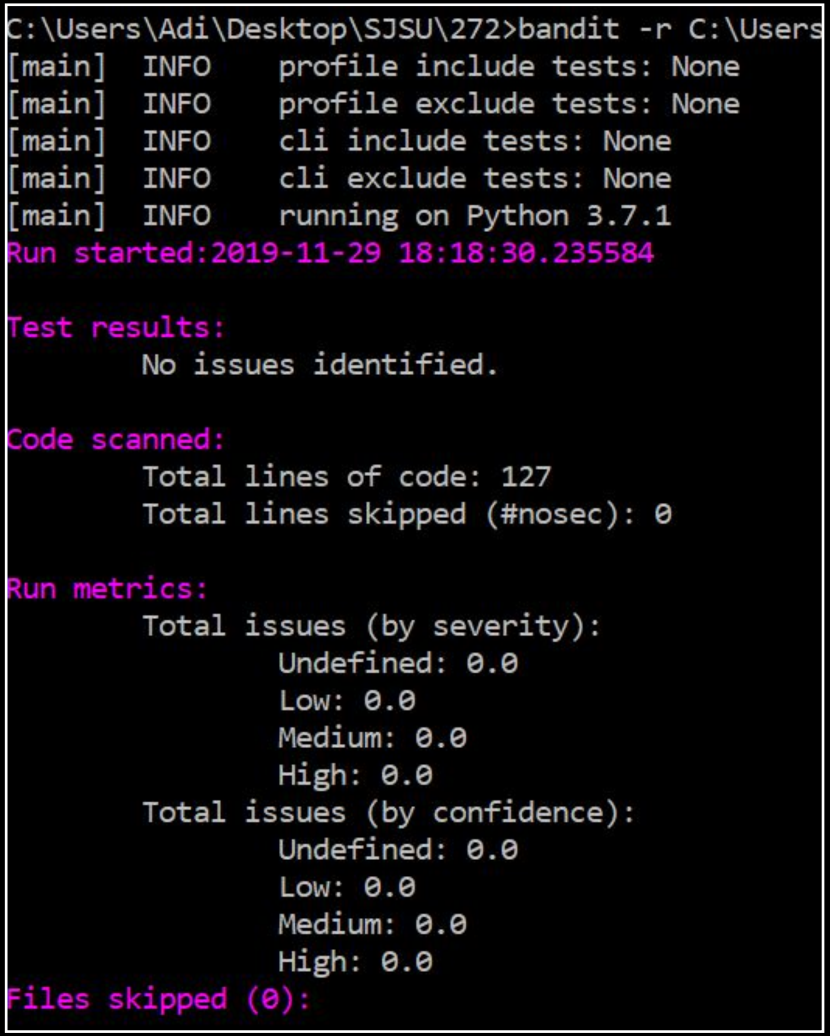
iOS

- React did not integrate with Play easily
- De-prioritized web app for iOS app
- Written in Swift
- Frameworks: Charts, MapKit, UIKit

Security

- Postgres credentials stored via AWS Secrets Manager
- Never committed to git

Code Quality



```
C:\Users\Adi\Desktop\SJSU\272>bandit -r C:\Users
[main]  INFO    profile include tests: None
[main]  INFO    profile exclude tests: None
[main]  INFO    cli include tests: None
[main]  INFO    cli exclude tests: None
[main]  INFO    running on Python 3.7.1
Run started:2019-11-29 18:18:30.235584

Test results:
        No issues identified.

Code scanned:
        Total lines of code: 127
        Total lines skipped (#nosec): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0.0
                Low: 0.0
                Medium: 0.0
                High: 0.0
        Total issues (by confidence):
                Undefined: 0.0
                Low: 0.0
                Medium: 0.0
                High: 0.0
Files skipped (0):
```

- Python: **Bandit** processes each file, builds an AST from it, and runs appropriate plugins against the AST nodes
- Scala: **Kiuwan** covers the most stringent security standards such as OWASP and CWE
- Swift: **SwiftFormat** for iOS application code

Q&A

References

- Cost vs. Value: https://www.remodeling.hw.net/cost-vs-value/2019
- tabula-java: https://github.com/tabulapdf/tabula-java
- scala-csv: https://www.github.com/tototoshi/scala-csv
- AWS Secrets Manager: https://docs.aws.amazon.com/secretsmanager/latest/userguide/manage_create-basic-secret.html
- AWS Secrets Tutorial: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/specifying-sensitive-data-tutorial.html
- Specifying Sensitive Data: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/specifying-sensitive-data.html
- Deploy Docker Containers with ECS: https://aws.amazon.com/getting-started/tutorials/deploy-docker-containers/
- Create and Connect to a PostgreSQL Database: https://aws.amazon.com/getting-started/tutorials/create-connect-postgresql-db/
- Setting Up with Amazon ECS: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/get-set-up-for-amazon-ecs.html
- CockroachDB: https://docs.aws.amazon.com/eks/latest/userguide/getting-started-eksctl.html
- Deploying CockroachDB to Amazon EKS: https://www.cockroachlabs.com/docs/v19.2/orchestrate-cockroachdb-with-kubernetes.html#step-7-maintain-the-cluster