SUGGESTING AFFORDABLE YET SAFE HOUSING IN SEATTLE, WA

BY: DR YURY CHEBIRYAK

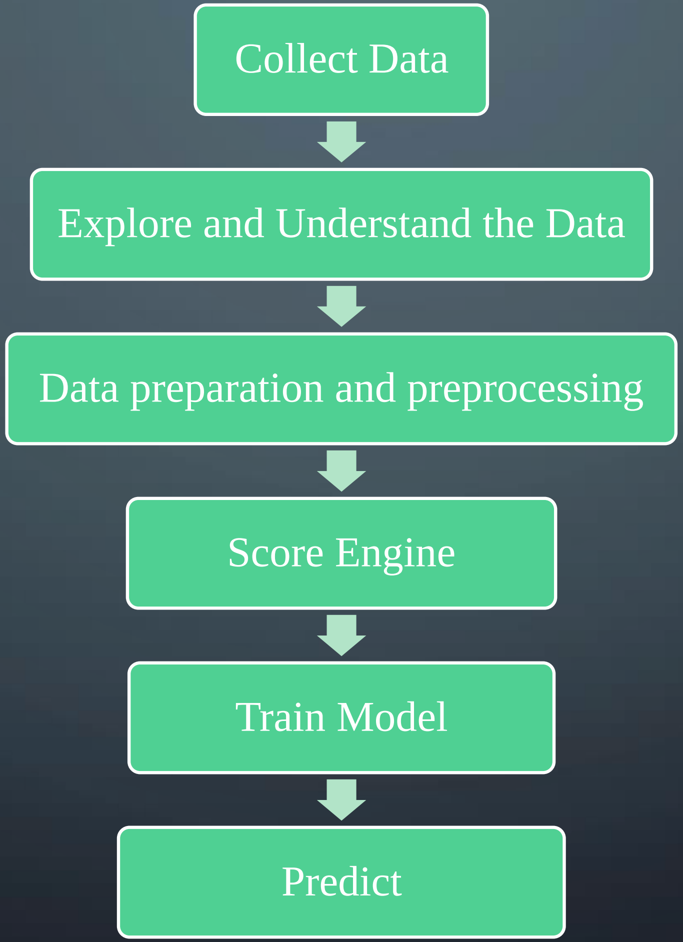
INTRODUCTION

- Athlete club
  - Several flats in Seattle, WA
  - Proximity to a park
  - Low criminality
  - Affordable
  - Extensible analysis to attract other potential clients: property buyers, etc

DATA DESCRIPTION

- Seattle neighborhoods compared using attributes
    - Criminality: use official sources
    - Location: crawl Wikipedia
    - Pricing: access Airbnb database
    - Proximity to parks: query Foursquare API
    - Population size: FindMySeattle portal pages

METHODOLOGY



Collect Data

Explore and Understand the Data

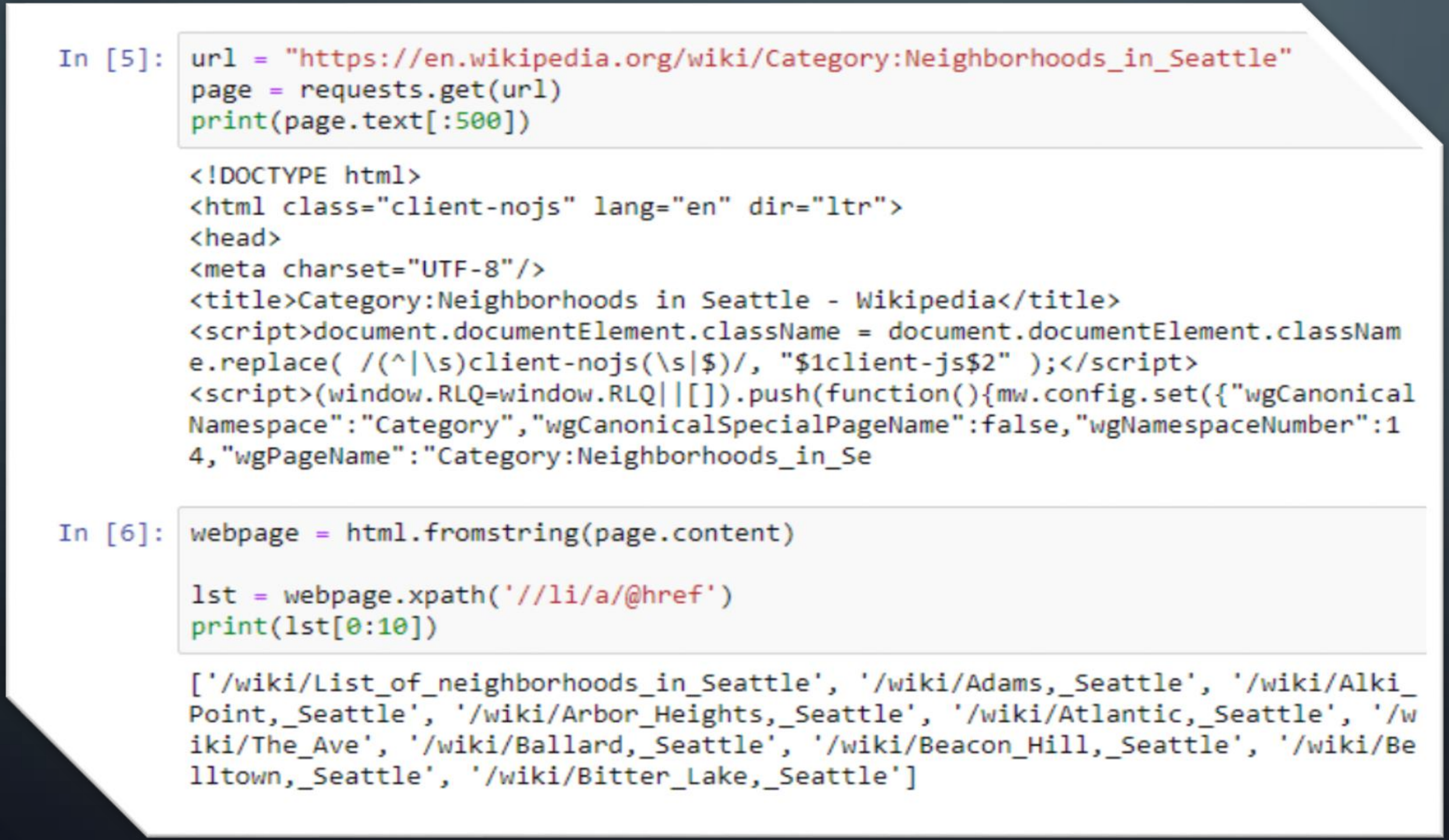
Data preparation and preprocessing

Score Engine

Train Model

Predict

OBTAIN LOCATIONS OF DISTRICTS



In [5]:

```
url = "https://en.wikipedia.org/wiki/Category:Neighborhoods_in_Seattle"
page = requests.get(url)
print(page.text[:500])
```

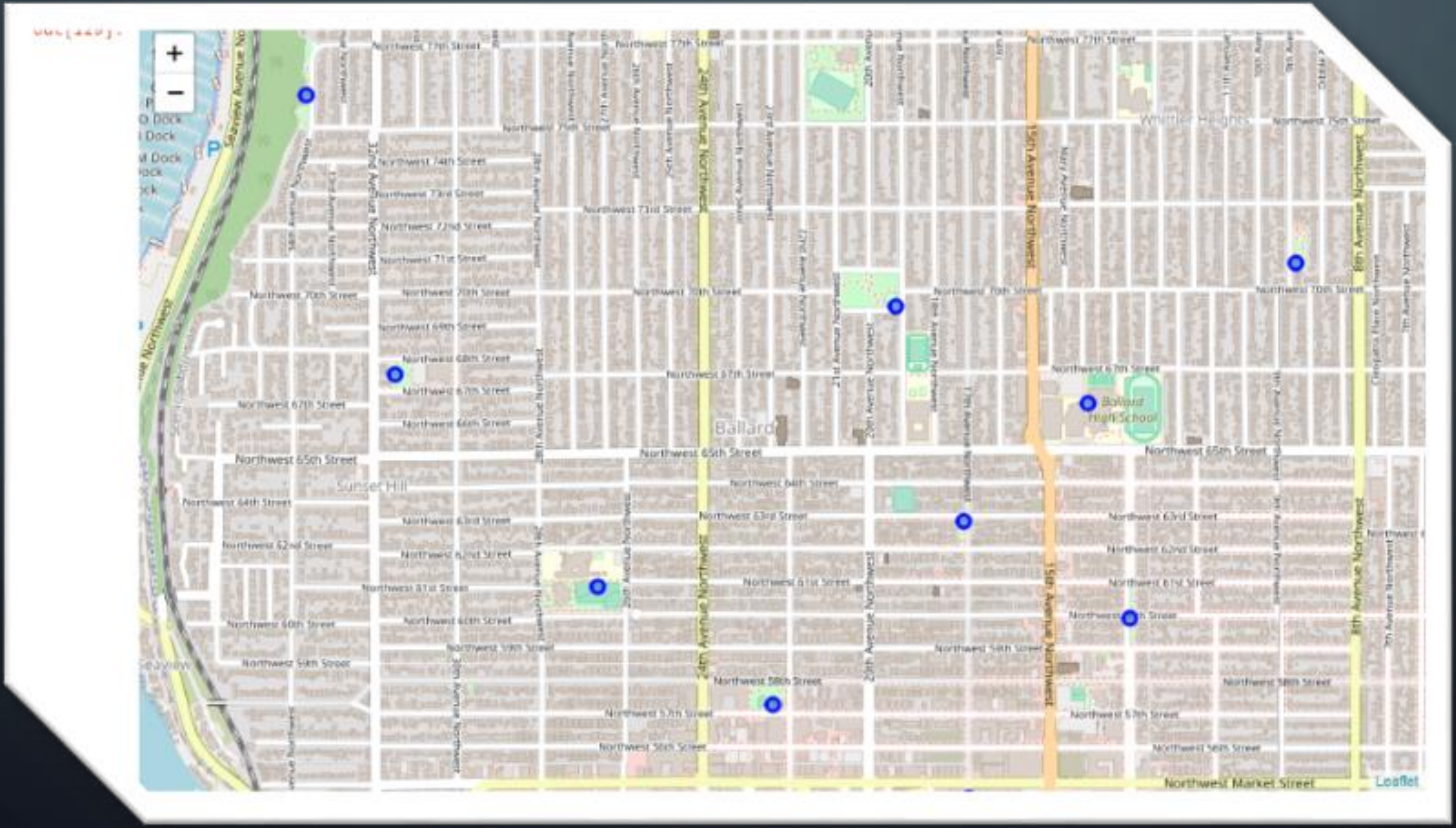
```
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>Category:Neighborhoods in Seattle - Wikipedia</title>
<script>document.documentElement.className = document.documentElement.classNam
e.replace( /(^|\s)client-nojs(\s|$)/, "$1client-js$2" );</script>
<script>(window.RLQ=window.RLQ||[]).push(function(){mw.config.set({"wgCanonical
Namespace":"Category","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":1
4,"wgPageName":"Category:Neighborhoods_in_Se
```

In [6]:

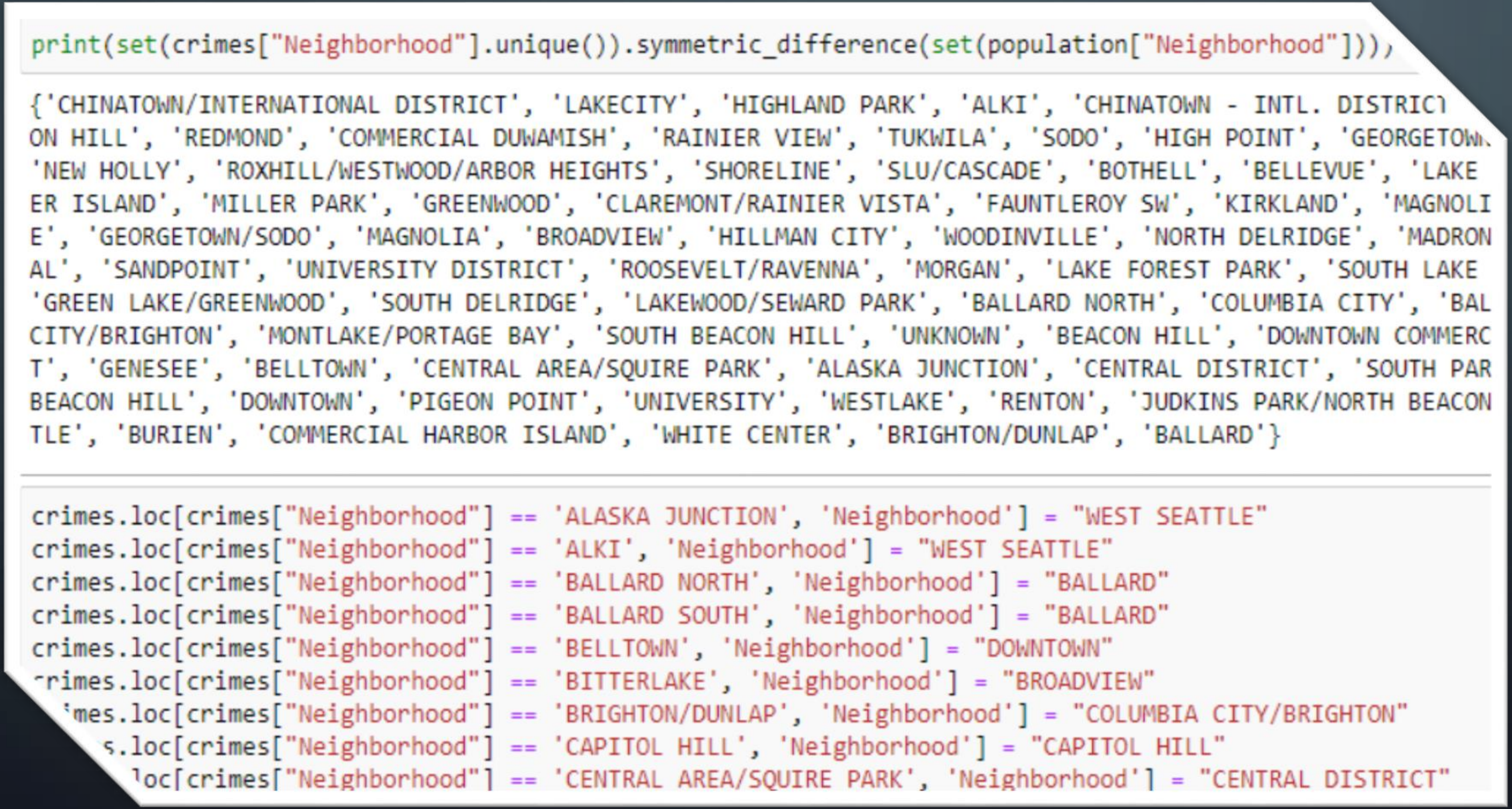
```
webpage = html.fromstring(page.content)

lst = webpage.xpath('//li/a/@href')
print(lst[0:10])
```

```
['/wiki/List_of_neighborhoods_in_Seattle', '/wiki/Adams,_Seattle', '/wiki/Alki_
Point,_Seattle', '/wiki/Arbor_Heights,_Seattle', '/wiki/Atlantic,_Seattle', '/w
iki/The_Ave', '/wiki/Ballard,_Seattle', '/wiki/Beacon_Hill,_Seattle', '/wiki/Be
lltown,_Seattle', '/wiki/Bitter_Lake,_Seattle']
```

VISUALIZE THE LOCATIONS OF PARKS



CLEAN AND NORMALIZE DISTRICT NAMES

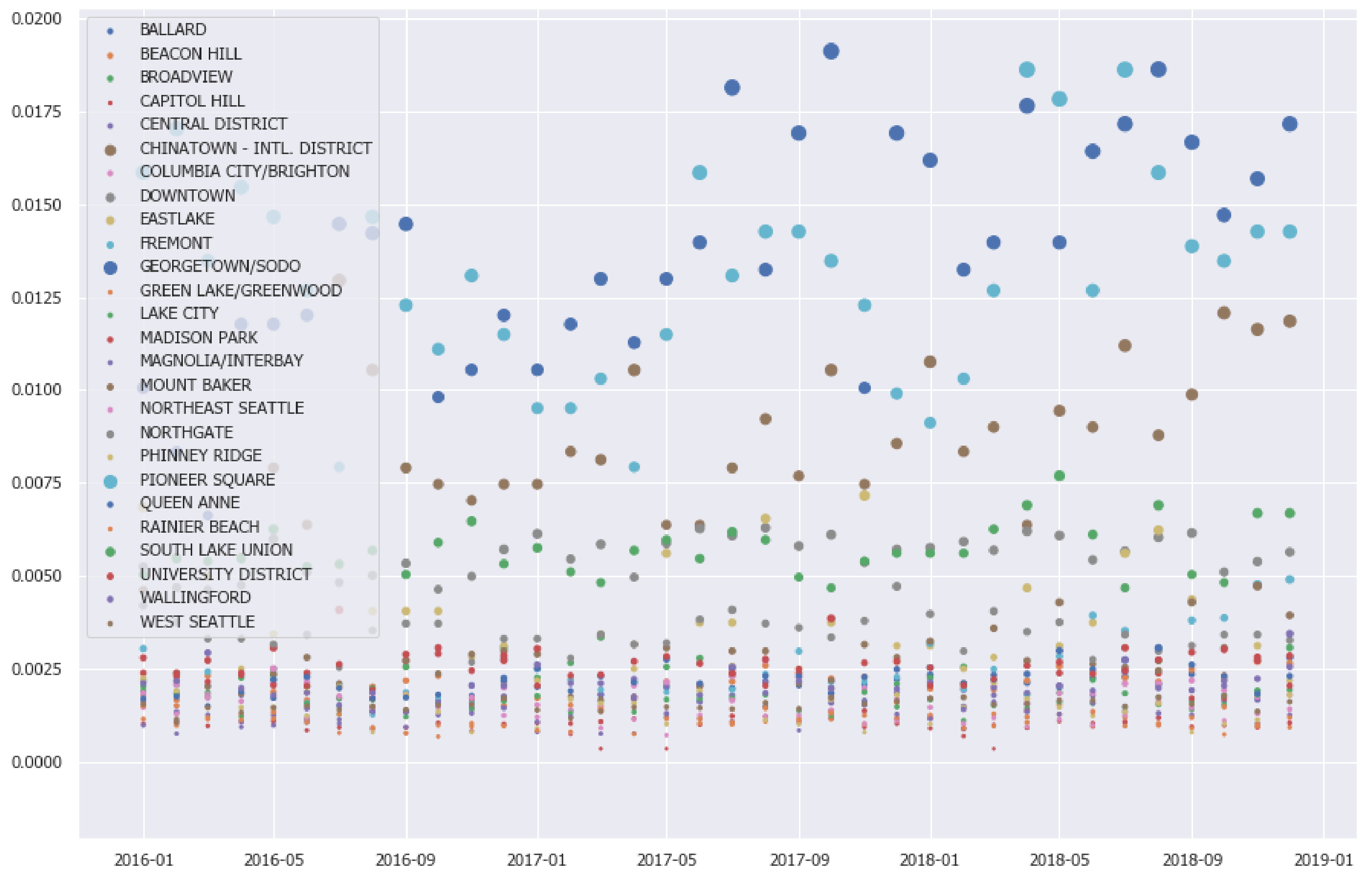


```
print(set(crimes["Neighborhood"].unique()).symmetric_difference(set(population["Neighborhood"])))
```

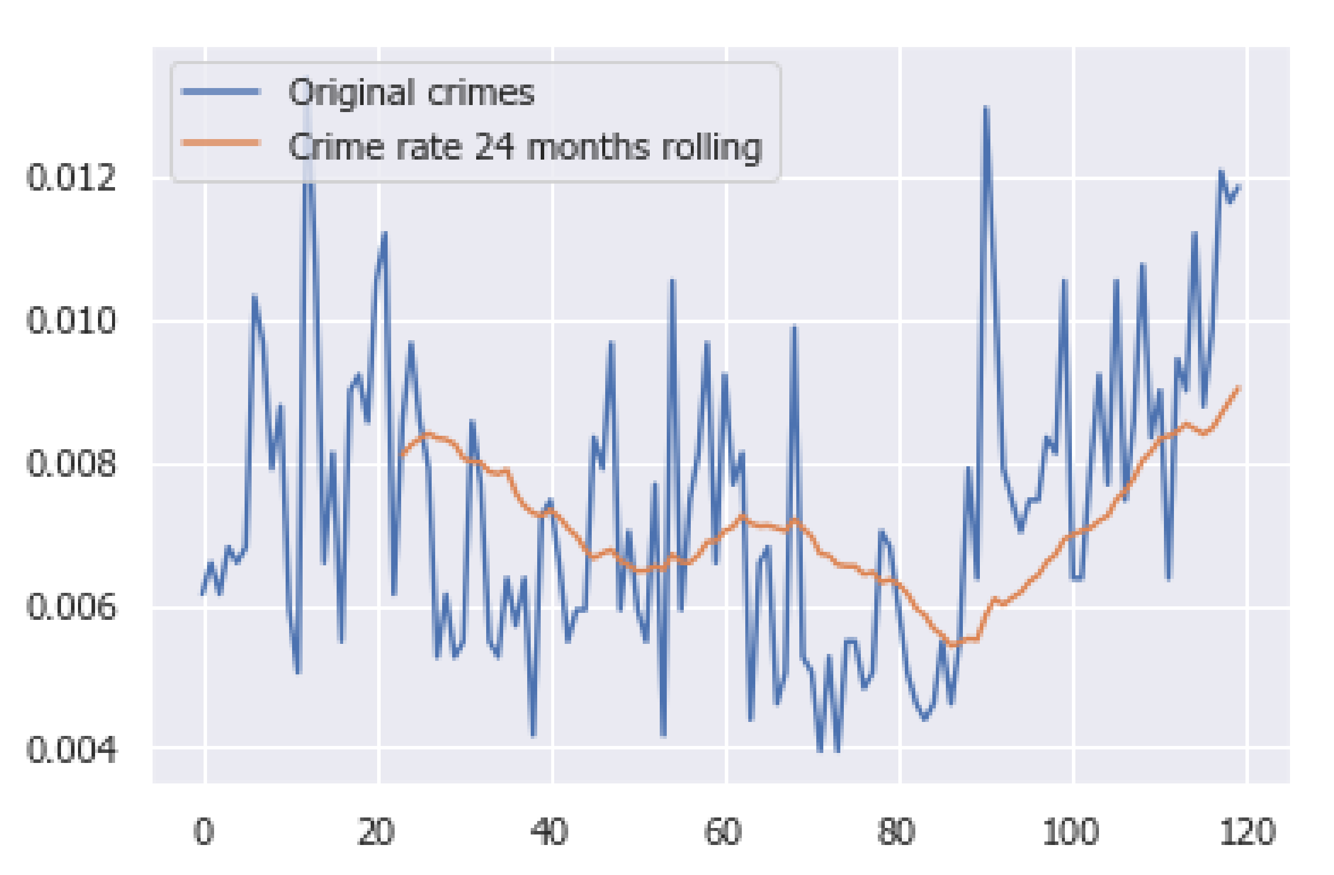
```
{'CHINATOWN/INTERNATIONAL DISTRICT', 'LAKECITY', 'HIGHLAND PARK', 'ALKI', 'CHINATOWN - INTL. DISTRICT
ON HILL', 'REDMOND', 'COMMERCIAL DUWAMISH', 'RAINIER VIEW', 'TUKWILA', 'SODO', 'HIGH POINT', 'GEORGETOWN
'NEW HOLLY', 'ROXHILL/WESTWOOD/ARBOR HEIGHTS', 'SHORELINE', 'SLU/CASCADE', 'BOTHELL', 'BELLEVUE', 'LAKE
ER ISLAND', 'MILLER PARK', 'GREENWOOD', 'CLAREMONT/RAINIER VISTA', 'FAUNTLEROY SW', 'KIRKLAND', 'MAGNOLI
E', 'GEORGETOWN/SODO', 'MAGNOLIA', 'BROADVIEW', 'HILLMAN CITY', 'WOODINVILLE', 'NORTH DELRIDGE', 'MADRON
AL', 'SANDPOINT', 'UNIVERSITY DISTRICT', 'ROOSEVELT/RAVENNA', 'MORGAN', 'LAKE FOREST PARK', 'SOUTH LAKE
'GREEN LAKE/GREENWOOD', 'SOUTH DELRIDGE', 'LAKEWOOD/SEWARD PARK', 'BALLARD NORTH', 'COLUMBIA CITY', 'BAL
CITY/BRIGHTON', 'MONTLAKE/PORTAGE BAY', 'SOUTH BEACON HILL', 'UNKNOWN', 'BEACON HILL', 'DOWNTOWN COMMERC
T', 'GENESEE', 'BELLTOWN', 'CENTRAL AREA/SQUIRE PARK', 'ALASKA JUNCTION', 'CENTRAL DISTRICT', 'SOUTH PAR
BEACON HILL', 'DOWNTOWN', 'PIGEON POINT', 'UNIVERSITY', 'WESTLAKE', 'RENTON', 'JUDKINS PARK/NORTH BEACON
TLE', 'BURIEN', 'COMMERCIAL HARBOR ISLAND', 'WHITE CENTER', 'BRIGHTON/DUNLAP', 'BALLARD'}
```

```
crimes.loc[crimes["Neighborhood"] == 'ALASKA JUNCTION', 'Neighborhood'] = "WEST SEATTLE"
crimes.loc[crimes["Neighborhood"] == 'ALKI', 'Neighborhood'] = "WEST SEATTLE"
crimes.loc[crimes["Neighborhood"] == 'BALLARD NORTH', 'Neighborhood'] = "BALLARD"
crimes.loc[crimes["Neighborhood"] == 'BALLARD SOUTH', 'Neighborhood'] = "BALLARD"
crimes.loc[crimes["Neighborhood"] == 'BELLTOWN', 'Neighborhood'] = "DOWNTOWN"
rimes.loc[crimes["Neighborhood"] == 'BITTERLAKE', 'Neighborhood'] = "BROADVIEW"
 mes.loc[crimes["Neighborhood"] == 'BRIGHTON/DUNLAP', 'Neighborhood'] = "COLUMBIA CITY/BRIGHTON"
   s.loc[crimes["Neighborhood"] == 'CAPITOL HILL', 'Neighborhood'] = "CAPITOL HILL"
     loc[crimes["Neighborhood"] == 'CENTRAL AREA/SQUIRE PARK', 'Neighborhood'] = "CENTRAL DISTRICT"
```

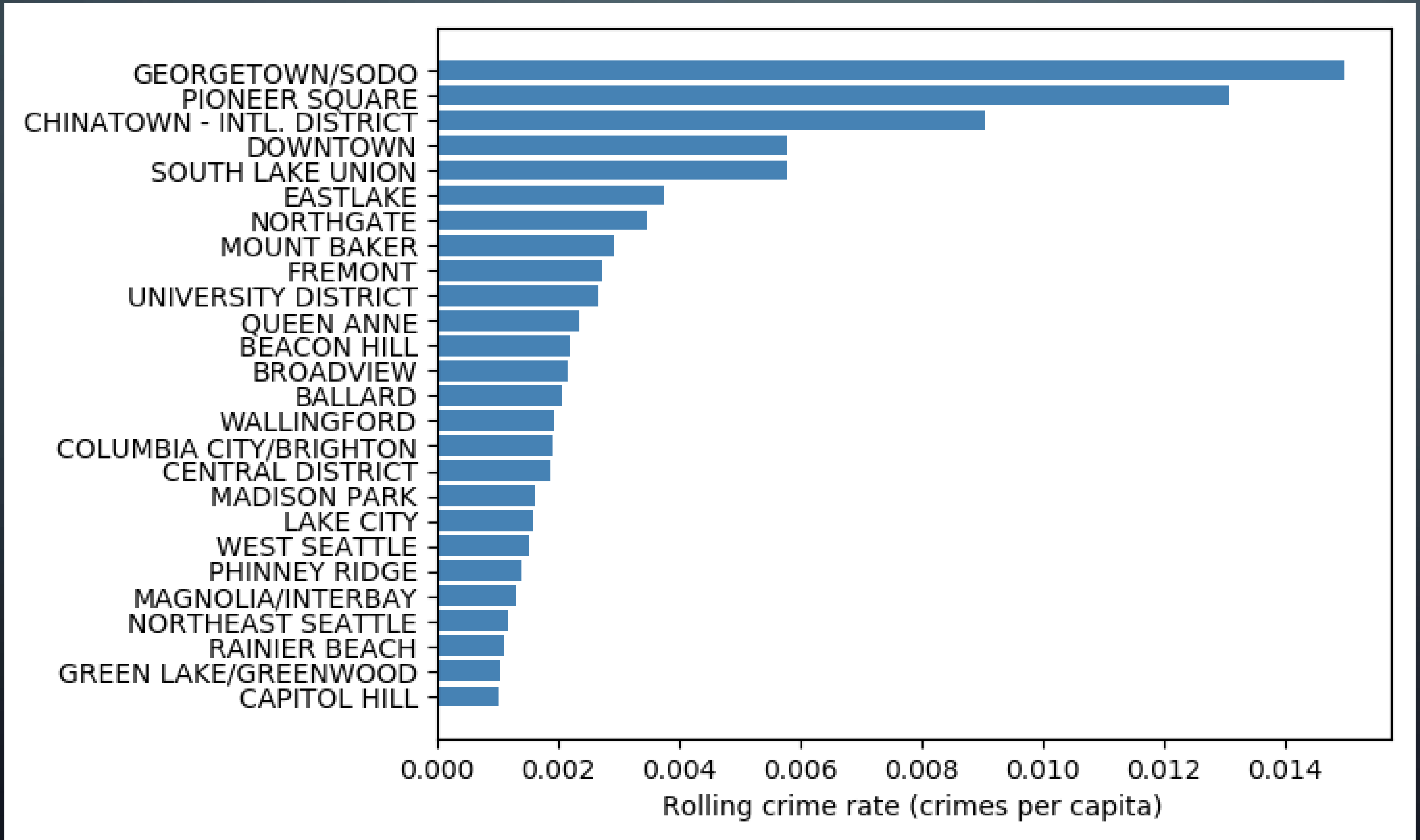
FORM MONTHLY BREAKDOWN OF CRIMES



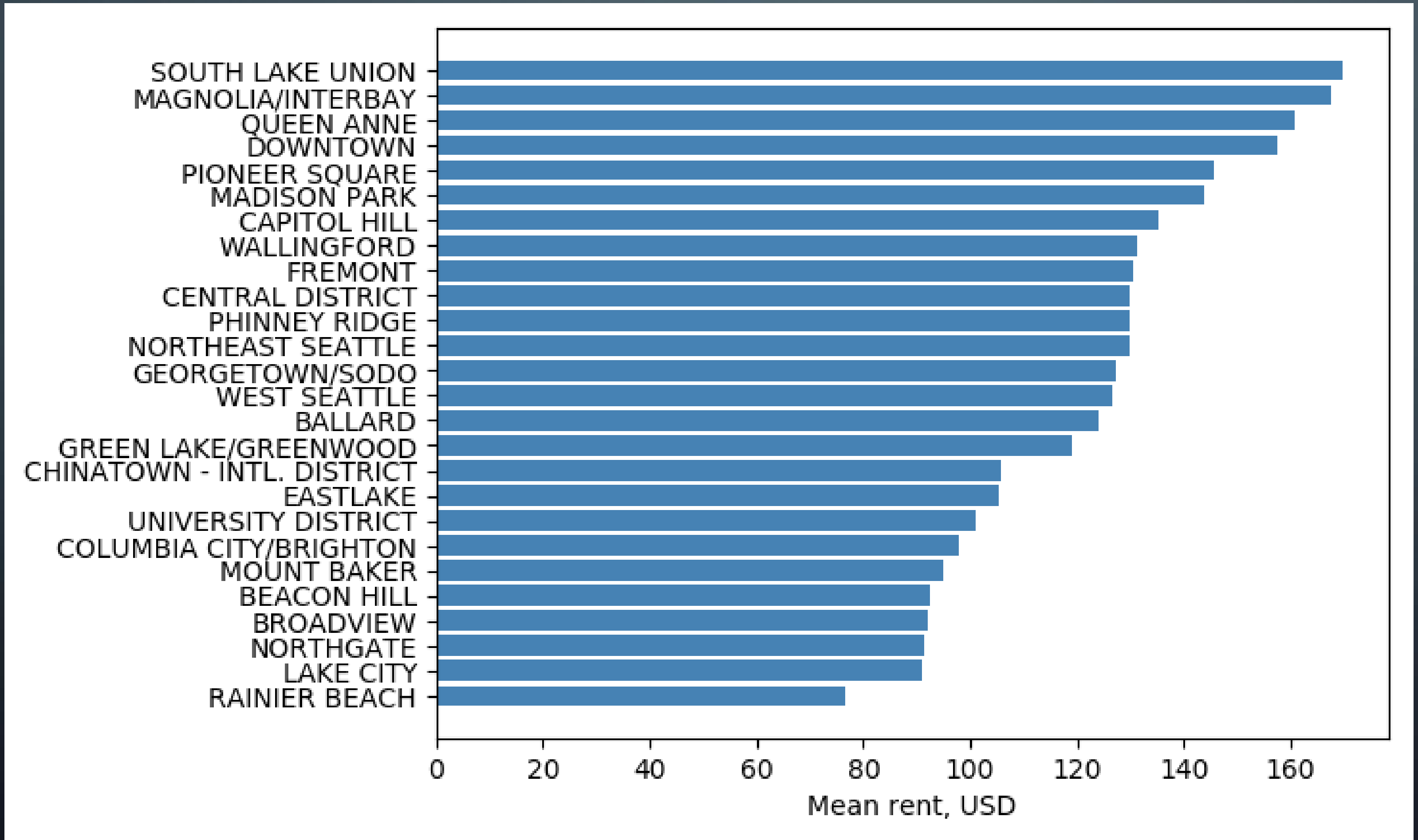
USE TWO-YEAR ROLLING PER CAPITA AS ATTRIBUTE



USE TWO-YEAR ROLLING PER CAPITA AS ATTRIBUTE



MEAN PRICE OF FLATS AS ATTRIBUTE



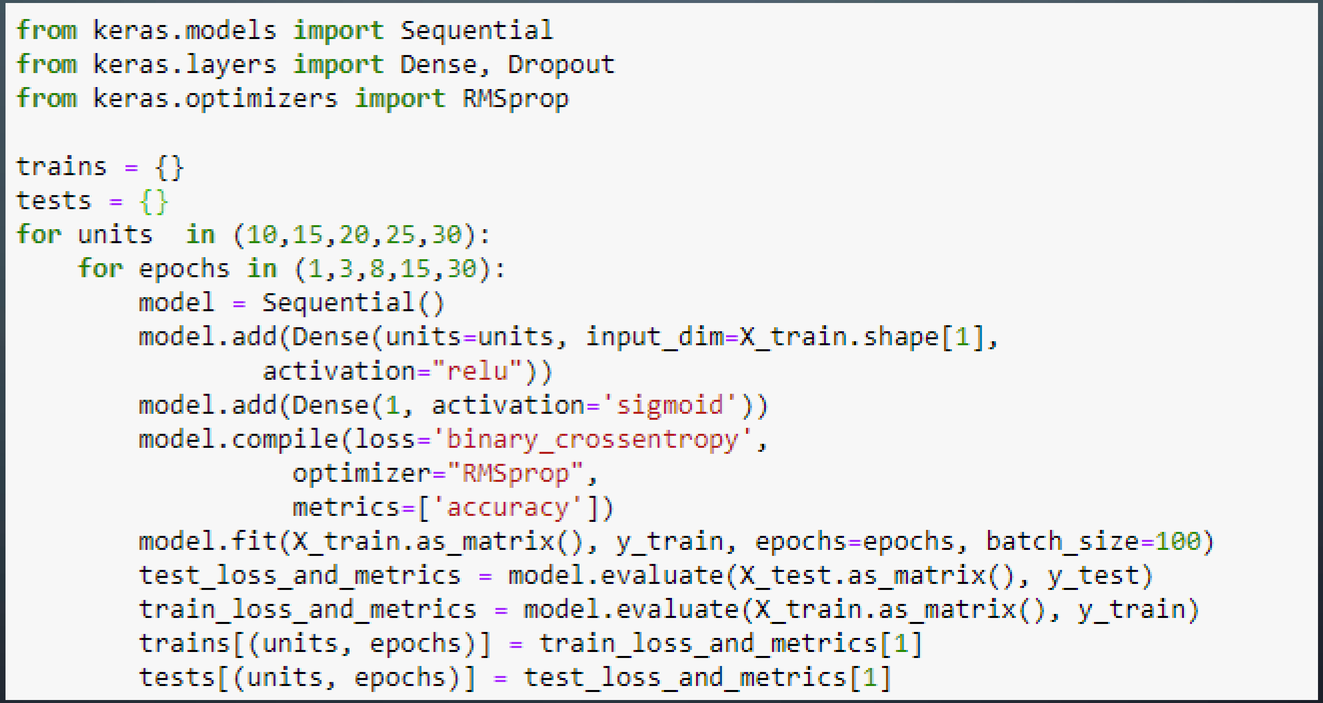
SCORING ENGINE

- ‘Desired’ flat if:
  - ParkScore + PriceScore * 1.25 + CriminalityScore * 1.5 >= 240, and
  - ParkScore >= 50
  - PriceScore >= 50
  - CriminalityScore >= 60
- Where each score is a percentile of their attribute’s distribution

SUPERVISED LEARNING

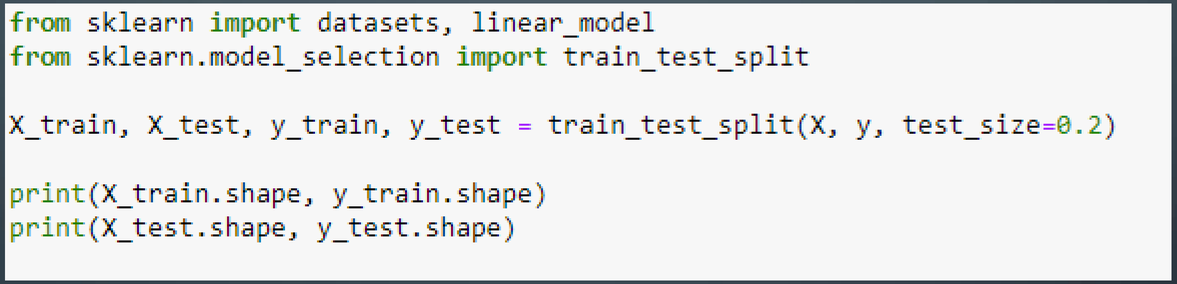
- Train a model to identify ‘desired’ flats
- Mimics content-based filtering with a single user profile

MODEL DEFINITION



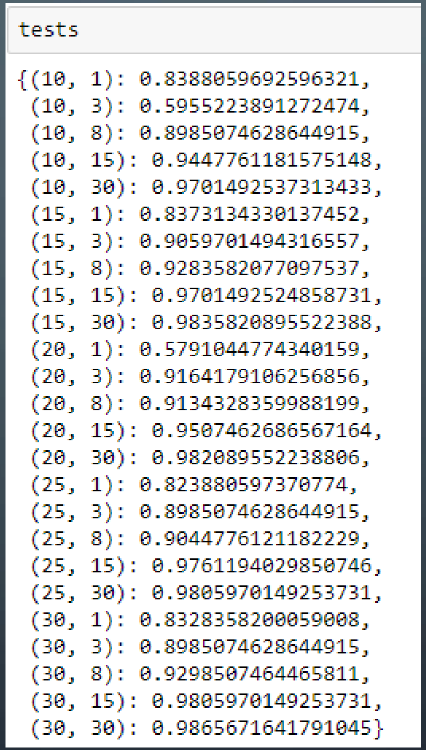
```python
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import RMSprop

trains = {}
tests = {}
for units  in (10,15,20,25,30):
    for epochs in (1,3,8,15,30):
        model = Sequential()
        model.add(Dense(units=units, input_dim=X_train.shape[1],
                activation="relu"))
        model.add(Dense(1, activation='sigmoid'))
        model.compile(loss='binary_crossentropy',
                  optimizer="RMSprop",
                  metrics=['accuracy'])
        model.fit(X_train.as_matrix(), y_train, epochs=epochs, batch_size=100)
        test_loss_and_metrics = model.evaluate(X_test.as_matrix(), y_test)
        train_loss_and_metrics = model.evaluate(X_train.as_matrix(), y_train)
        trains[(units, epochs)] = train_loss_and_metrics[1]
        tests[(units, epochs)] = test_loss_and_metrics[1]
```

MODEL TRAINING



```
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

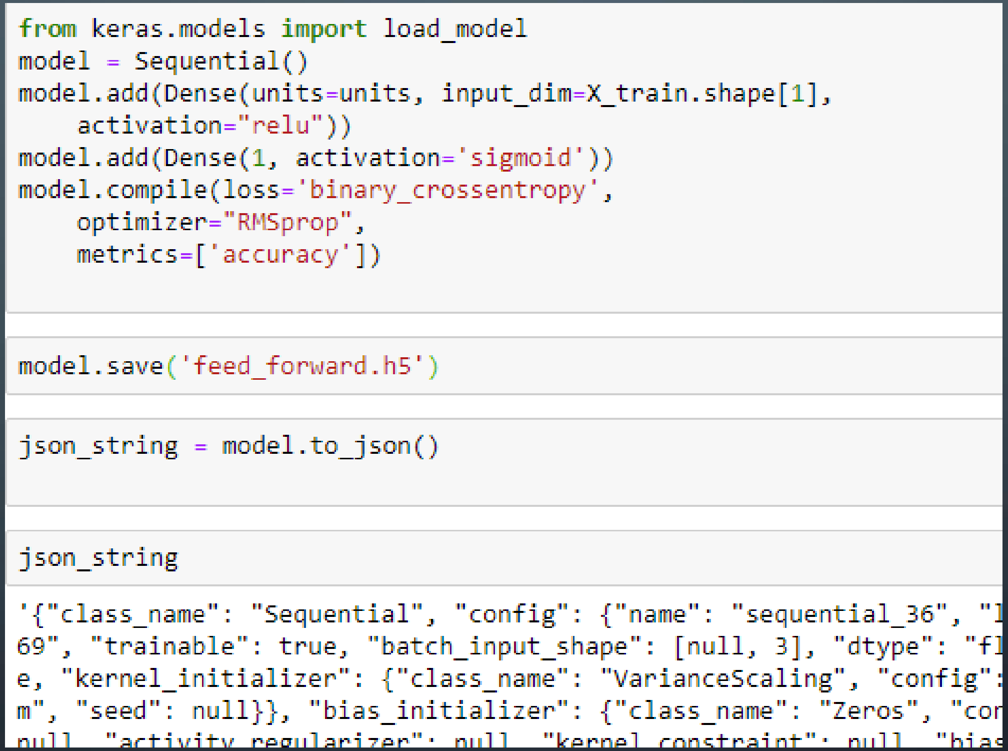
MODEL EVALUATION



```
tests
```

```
{(10, 1): 0.8388059692596321,
 (10, 3): 0.5955223891272474,
 (10, 8): 0.8985074628644915,
 (10, 15): 0.9447761181575148,
 (10, 30): 0.9701492537313433,
 (15, 1): 0.8373134330137452,
 (15, 3): 0.9059701494316557,
 (15, 8): 0.9283582077097537,
 (15, 15): 0.9701492524858731,
 (15, 30): 0.9835820895522388,
 (20, 1): 0.5791044774340159,
 (20, 3): 0.9164179106256856,
 (20, 8): 0.913432835998199,
 (20, 15): 0.9507462686567164,
 (20, 30): 0.982089552238806,
 (25, 1): 0.823880597370774,
 (25, 3): 0.8985074628644915,
 (25, 8): 0.9044776121182229,
 (25, 15): 0.9761194029850746,
 (25, 30): 0.9805970149253731,
 (30, 1): 0.8328358200059008,
 (30, 3): 0.8985074628644915,
 (30, 8): 0.9298507464465811,
 (30, 15): 0.9805970149253731,
 (30, 30): 0.9865671641791045}
```

MODEL DEPLOYMENT



```python
from keras.models import load_model
model = Sequential()
model.add(Dense(units=units, input_dim=X_train.shape[1],
    activation="relu"))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy',
    optimizer="RMSprop",
    metrics=['accuracy'])
```

```python
model.save('feed_forward.h5')
```

```python
json_string = model.to_json()
```

```python
json_string
```

```
'{"class_name": "Sequential", "config": {"name": "sequential_36", "l
69", "trainable": true, "batch_input_shape": [null, 3], "dtype": "fl
e, "kernel_initializer": {"class_name": "VarianceScaling", "config":
m", "seed": null}}, "bias_initializer": {"class_name": "Zeros", "cor
null, "activity_regularizer": null, "kernel_constraint": null, "bias
```