WhyExpat

- Our Project: Buzz or Bust
- Questionnaire: Data gathering
- Case Studies : Neural Network and Randon Forest

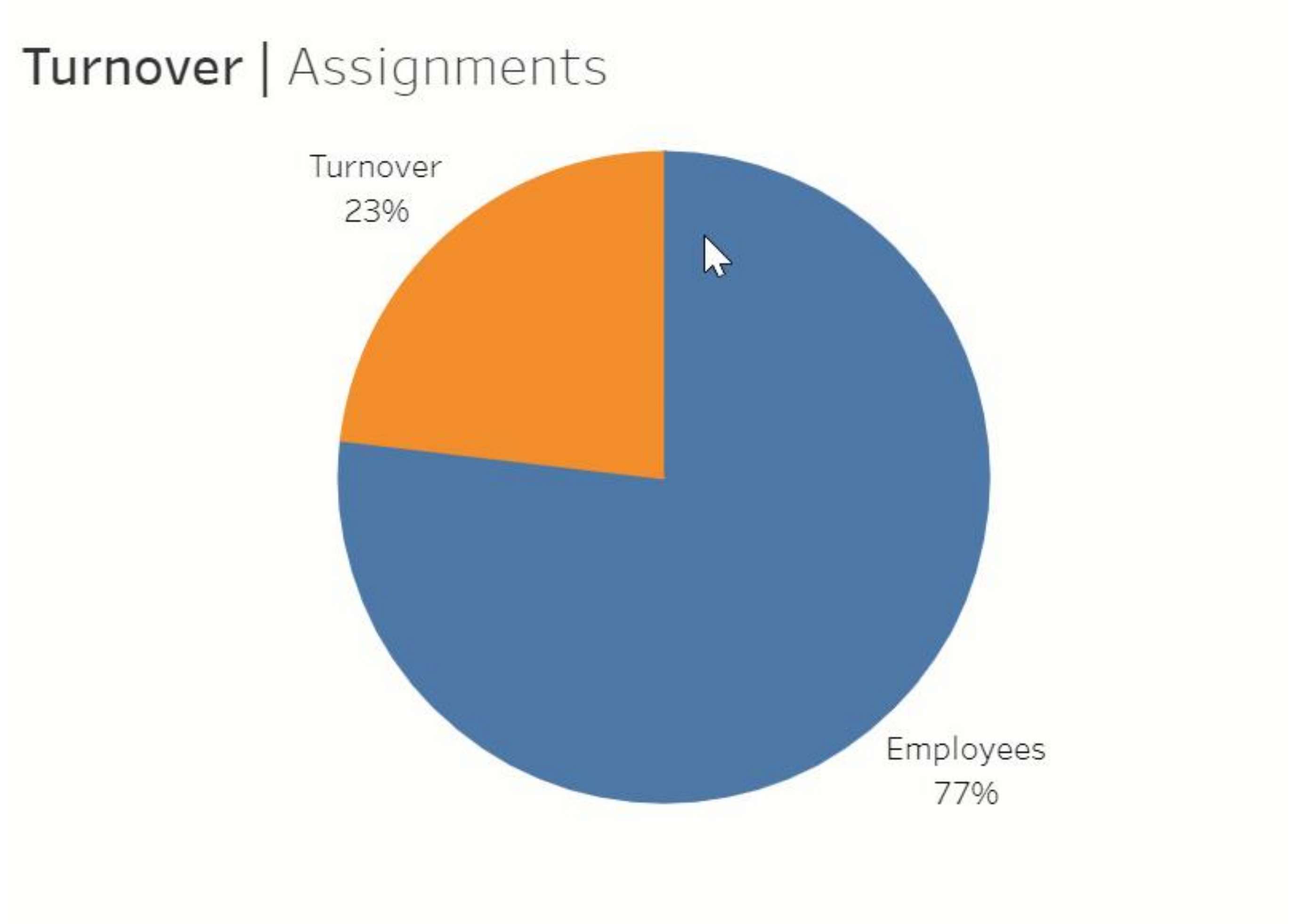
Repo: https://github.com/jvbonini/WhyExpat_site

Team Members: Lei Qin | Rob Seaberg | Winnie Wu | Joao Victor Bonini

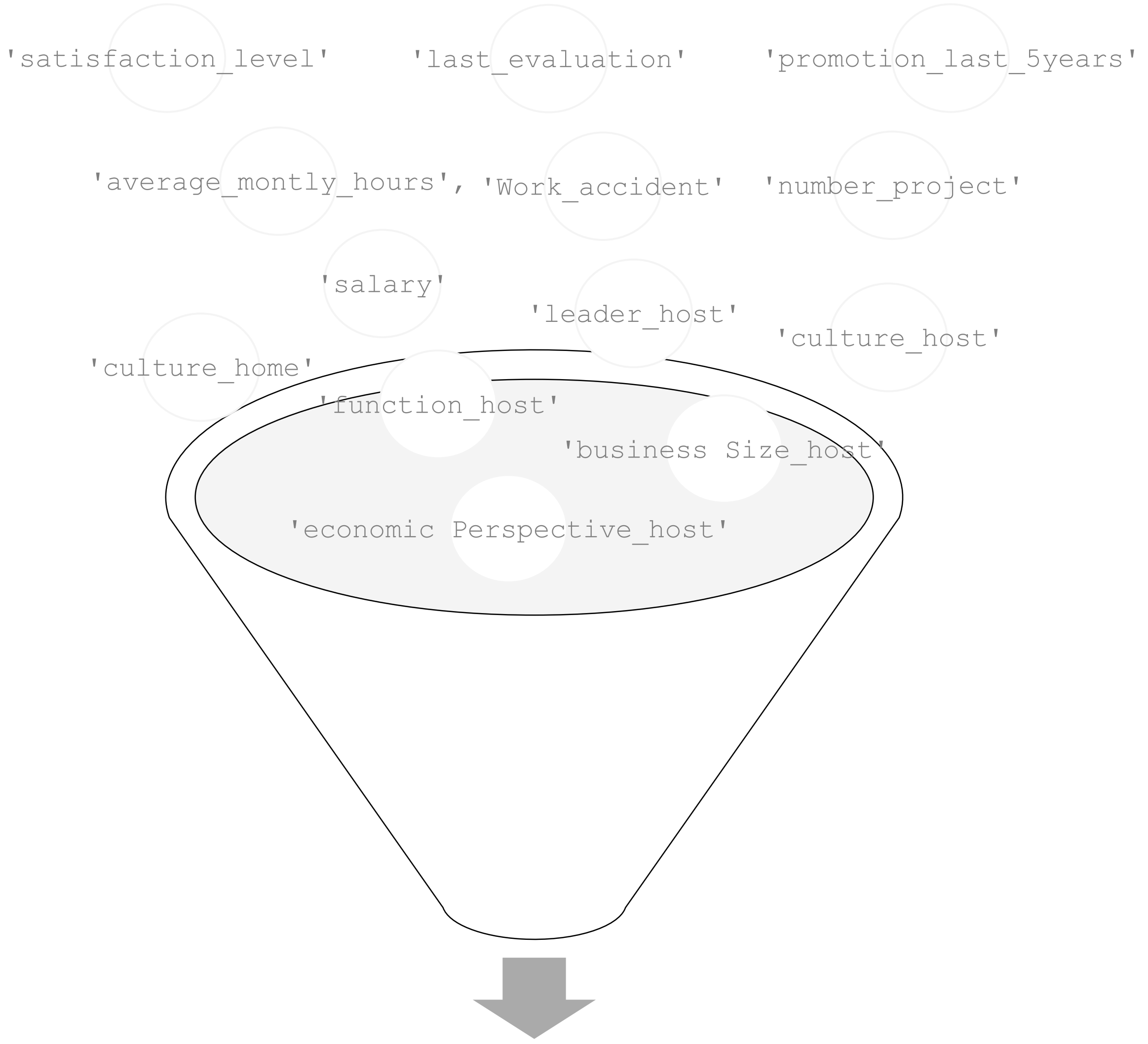
70 Million

www.WhyExpat.net

Our Project



Our Project



'satisfaction_level' 'last_evaluation' 'promotion_last_5years'

'average_montly_hours', 'Work_accident' 'number_project'

'salary'

'leader_host'

'culture_host'

'culture_home'

'function_host'

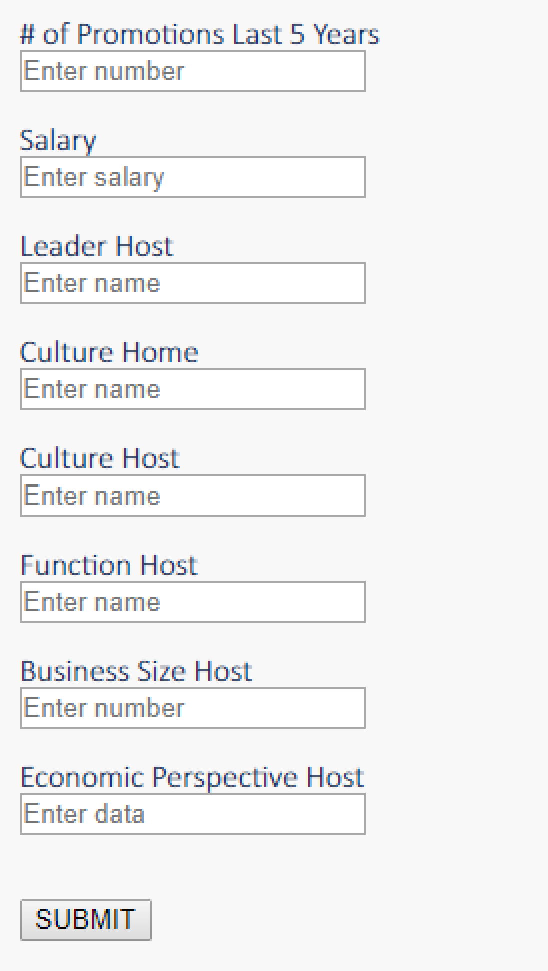
'business Size_host'

'economic Perspective_host'

'buzz_bust'

Questionnaire

Data gathering



# of Promotions Last 5 Years
Enter number

Salary
Enter salary

Leader Host
Enter name

Culture Home
Enter name

Culture Host
Enter name

Function Host
Enter name

Business Size Host
Enter number

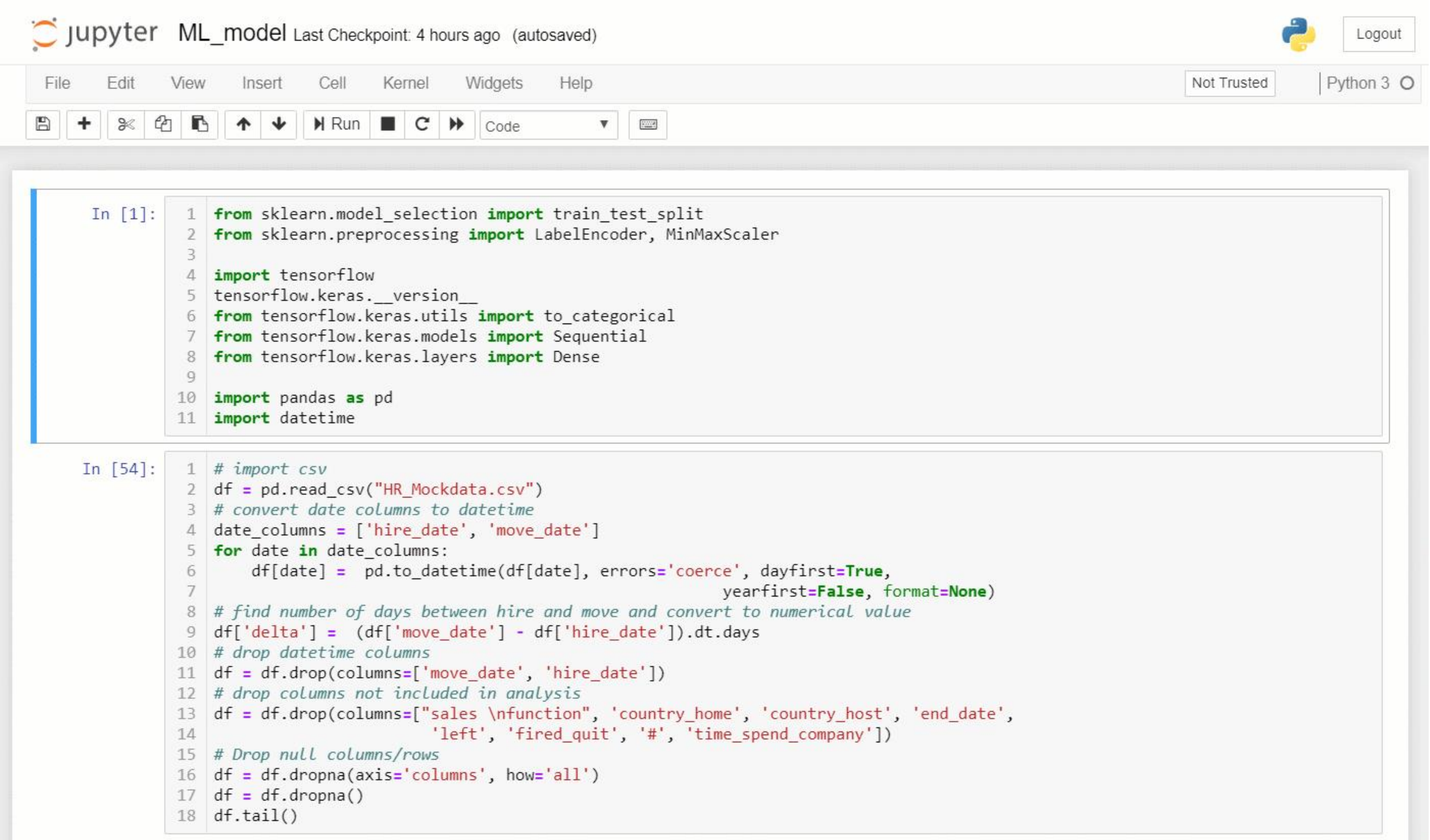
Economic Perspective Host
Enter data

Viewers perusing the site have an opportunity to fill out the questionnaire and add to the database

Data that is collected can be used to further hone the models and give stronger predictions

Case Studies

How might be used:



In [1]:

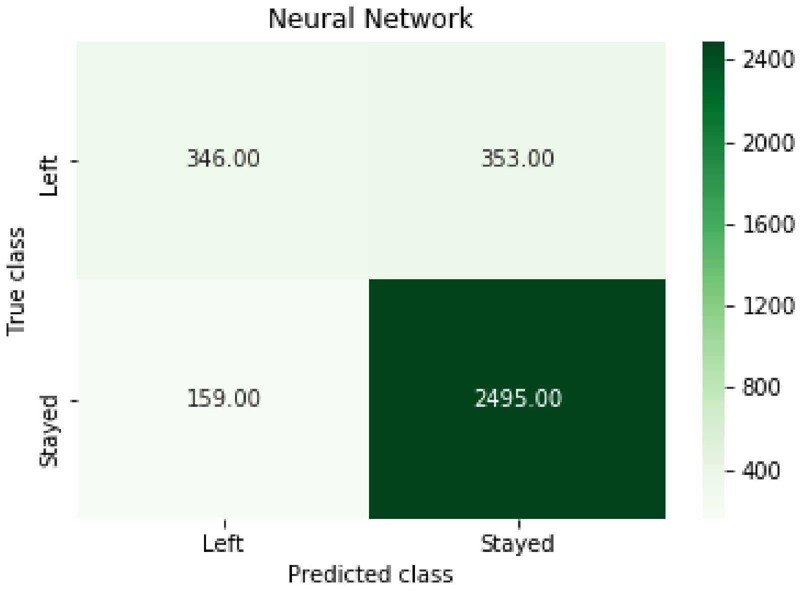
```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

import tensorflow
tensorflow.keras.__version__
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

import pandas as pd
import datetime
```

In [54]:

```python
# import csv
df = pd.read_csv("HR_Mockdata.csv")
# convert date columns to datetime
date_columns = ['hire_date', 'move_date']
for date in date_columns:
    df[date] =  pd.to_datetime(df[date], errors='coerce', dayfirst=True,
                                                      yearfirst=False, format=None)
# find number of days between hire and move and convert to numerical value
df['delta'] =  (df['move_date'] - df['hire_date']).dt.days
# drop datetime columns
df = df.drop(columns=['move_date', 'hire_date'])
# drop columns not included in analysis
df = df.drop(columns=["sales \nfunction", 'country_home', 'country_host', 'end_date',
                       'left', 'fired_quit', '#', 'time_spend_company'])
# Drop null columns/rows
df = df.dropna(axis='columns', how='all')
df = df.dropna()
df.tail()
```

Neural Network

- – Model accuracy: 0.847 (loss: 0.333)
- – Precision:



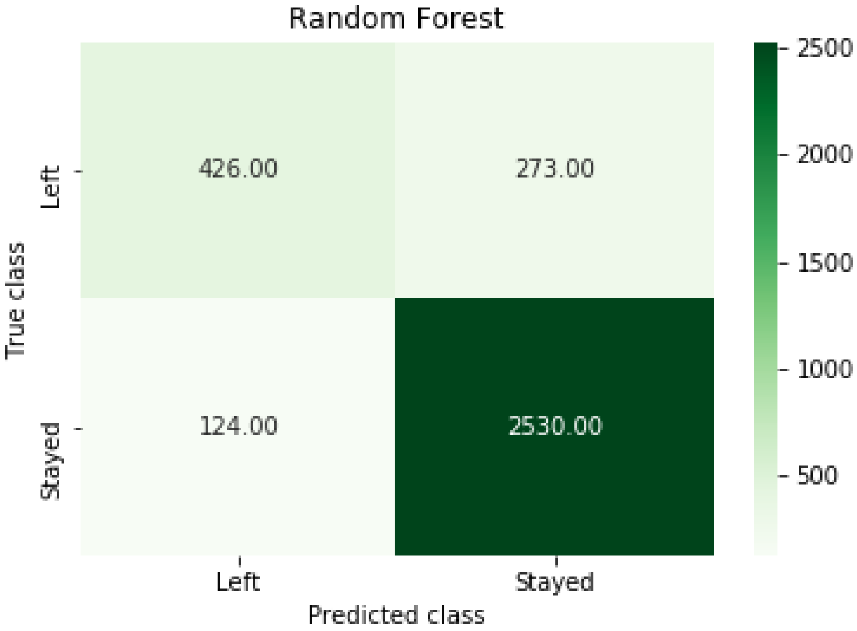
Retention rate:

79.1% (original)

88.0% (NN model)

Random Forest (Binary classification)

- Model accuracy: 0.853
- Precision:



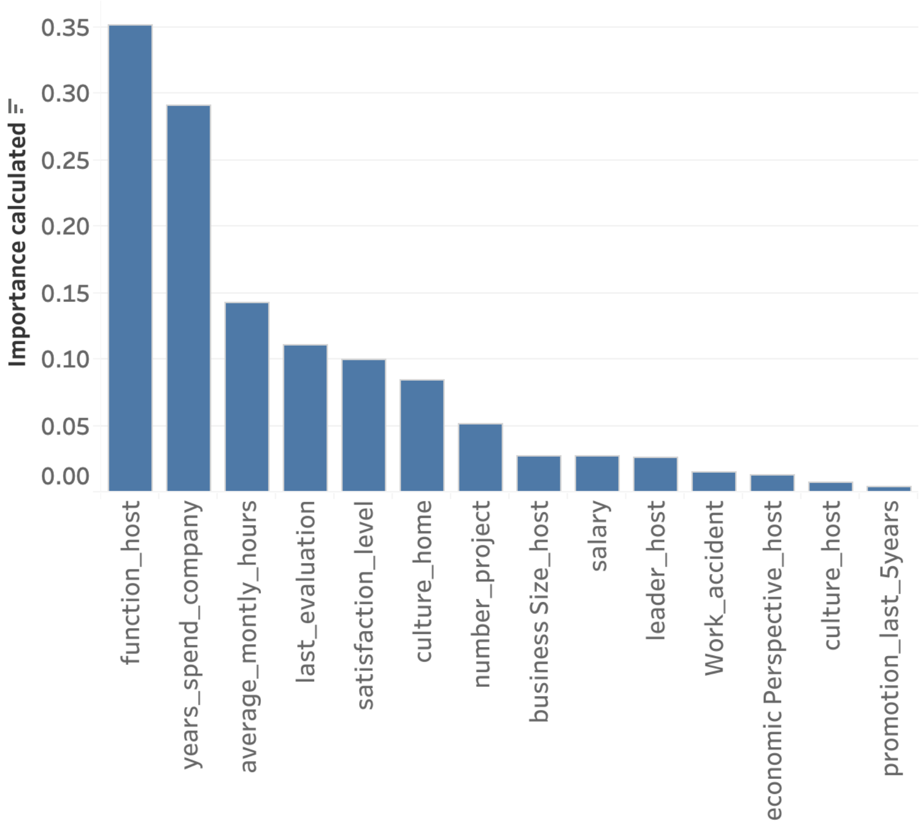
Retention rate:

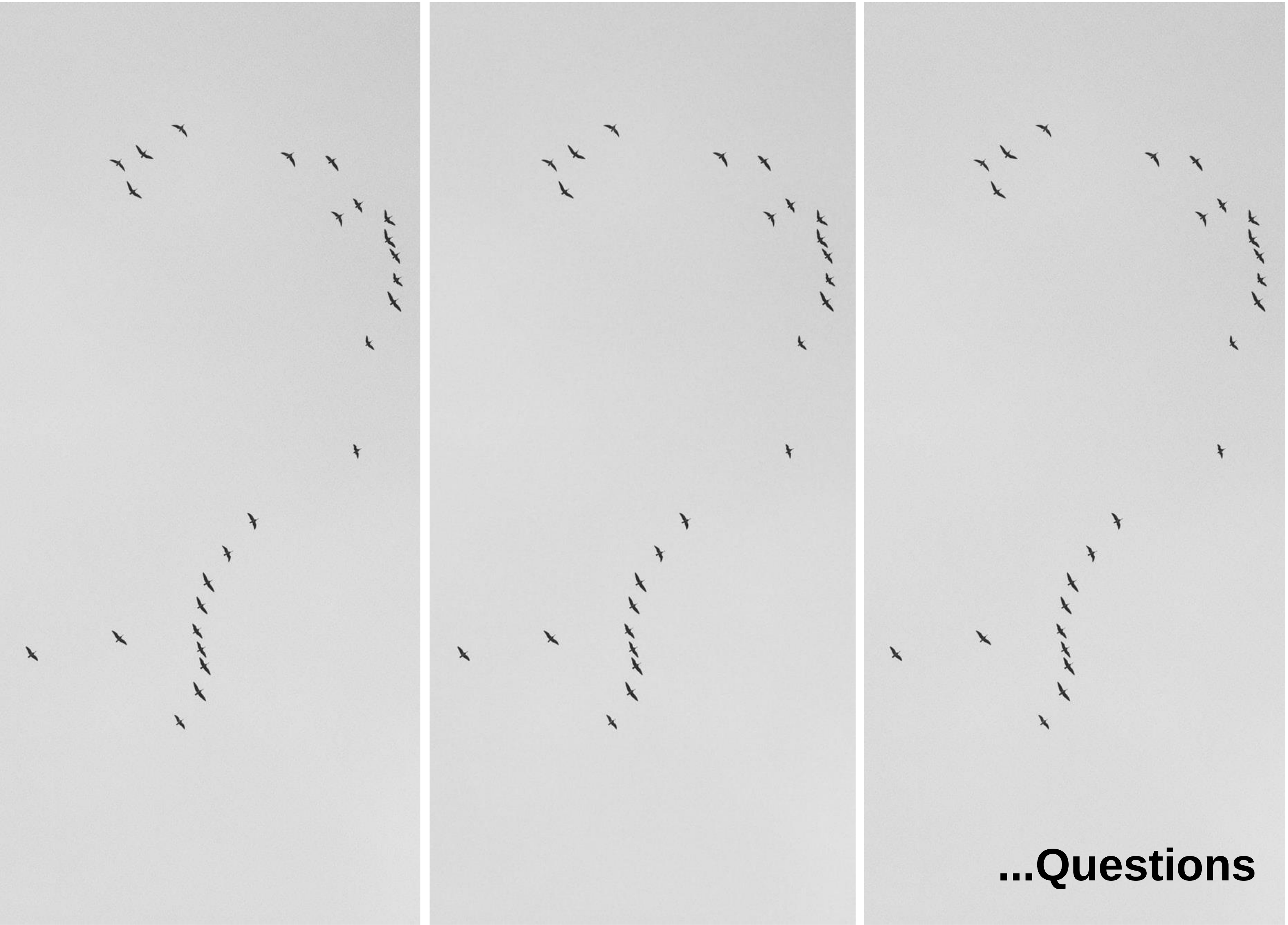
79.1% (original)

90.0% (RF model)

Random Forest

– Feature importance





...Questions

End | Lei Qin | Rob Seaberg | Winnie Wu |Joao Victor Bonini

www.WhyExpat.net