Data Science 101

Data Science Team

02/27/2020

Andrew Wheeler, PhD

andrew.wheeler@hms.com

Who we are – the Data Science team

Sanjeev Kumar

VP, AI, Data Engineering & Analytics

Bo Gu

Director of Data Science

Indu Govindasamy

Program Manager

Imad Dabbura

Data Scientist

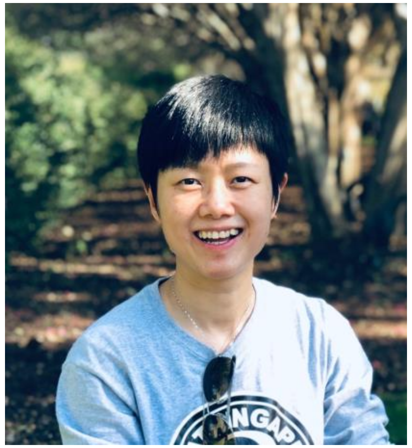
Puneet Girdhar

Data Scientist

Andrew Wheeler

Data Scientist

Yifei Yun



Data Scientist

Agenda

- What is Data Science?
- Data Science Workflow
- Brief Prediction Example in Python
- Future Topics & Questions

What is Data Science

- Using data to help people make better decisions:
  - Predictive modelling – identifying claims that have a high probability of match
  - Cost-Benefit analysis – knowing how many claims to audit that is cost-efficient
  - Experimental Evaluation – seeing if “strategy A” or “strategy B” results in more revenue
  - Automating routine/labor intensive tasks – instead of scanning 1000’s of claims, flagging a smaller number for review

- What it is not:
  - “Artificial Intelligence” (Skynet) – humans will always need to be involved in some capacity

Types of Data Science Problems

- Supervised learning, when we have historical data on the outcome of interest
  - Regression (predicting a continuous value, e.g. the amount of overpayment)
  - Classification (predicting the category, e.g. insurance should have paid claim)

- Unsupervised learning, trying to infer data that is not “labelled”
  - Text processing, e.g. seeing if documents have similar patterns
  - Merging unique identifiers across databases

- Reinforcement Learning

Typical Data Science Process Flow

Define Outcomes

- What you want predicted
- Criteria for Success

Data Steps

- Data Acquisition
- Data Cleaning
- Exploratory Data Analysis

Modelling Steps

- Feature Engineering (with Business Domain Knowledge)
- Create Predictive Models
- Benchmarking / Evaluation

Putting in Production

- Apply predictions to new cases
- Monitor Outcomes
- Deployment and Scalability

Example Using Python

- Majority of data science practitioners use the *Anaconda* distribution for python, https://www.anaconda.com/distribution/ (works on all operating systems)
  - Includes the majority of packages data scientists work with, along with an IDE (Spyder)
  - Jupyter notebooks are like interactive programming environments popular for data science

- What we will be doing today

  1) Load in data
  2) Browse Data & Create a graph
  3) Estimate a Regression Equation
  4) Apply predictions to new data

- Example predicting **obesity** using data from the Behavioral Risk Factor Survey
- Original data can be downloaded from https://health.data.ny.gov/Health/Behavioral-Risk-Factor-Surveillance-Survey-2015/rcr8-b3jj (I've only chosen a subset of variables.)

Exploratory Data Analysis (EDA)

In [1]:
```
#Loading in the libraries we will be using
import pandas as pd
from sklearn.linear_model import LogisticRegression
import os

#Setting the working directory to where our data is stored
os.chdir(r'C:\Users\e009156\Documents\DataScience_Notes\DataScience_101')

#Reading in the CSV data
brfss_dat = pd.read_csv('Prepped_BRFSS2015.csv')

#A quick view of the first few rows of data
brfss_dat.head()
```

Out[1]:

	Obese_BMI	CurrentSmoker	SEX	MinActWeek	AgeMid
0	1	0	Male	120.0	70
1	0	0	Female	0.0	60
2	0	0	Male	336.0	70
3	0	0	Female	420.0	30
4	0	0	Female	300.0	60

Can also import data directly from a SQL query

Numeric Data Stats

In [2]:
```
#Browsing the data

brfss_dat.describe()
```

Out[2]:

	Obese_BMI	CurrentSmoker	MinActWeek	AgeMid
count	11156.000000	11156.000000	11156.000000	11156.000000
mean	0.262011	0.134726	133.691466	54.147544
std	0.439749	0.341446	240.147264	15.486705
min	0.000000	0.000000	0.000000	20.000000
25%	0.000000	0.000000	0.000000	40.000000
50%	0.000000	0.000000	56.000000	60.000000
75%	1.000000	0.000000	180.000000	70.000000
max	1.000000	1.000000	3360.000000	70.000000

brfss_dat is our dataset object, which has various methods to plot and view the data

Categorical Data Stats

```
In [3]: #Can also look at the counts of individual categories

        brfss_dat['SEX'].value_counts()

Out[3]: Female    6280
        Male      4876
        Name: SEX, dtype: int64
```

dataframe[‘variable_name’]

selects a particular column of data

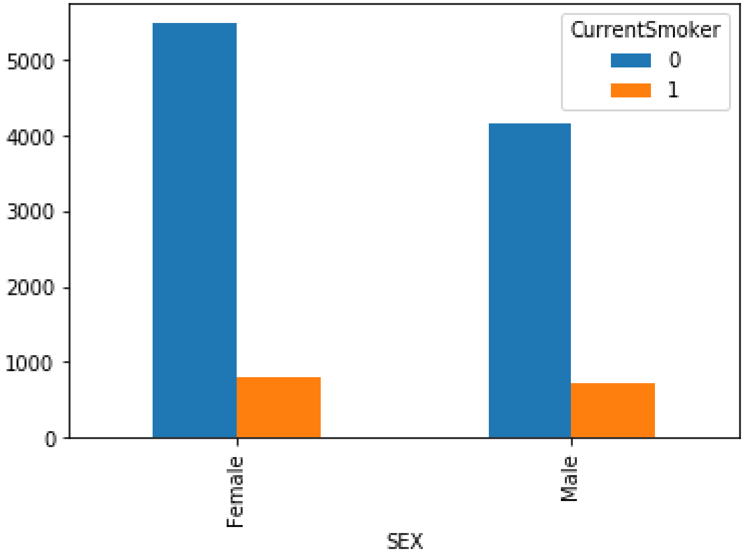
Data Visualization

In [9]:
```
#Sex by Smoking status

smoke_ct = pd.crosstab(brfss_dat['SEX'],brfss_dat['CurrentSmoker'])
smoke_ct.plot.bar()
smoke_ct
```

Out[9]:

CurrentSmoker	0	1
SEX		
Female	5487	793
Male	4166	710



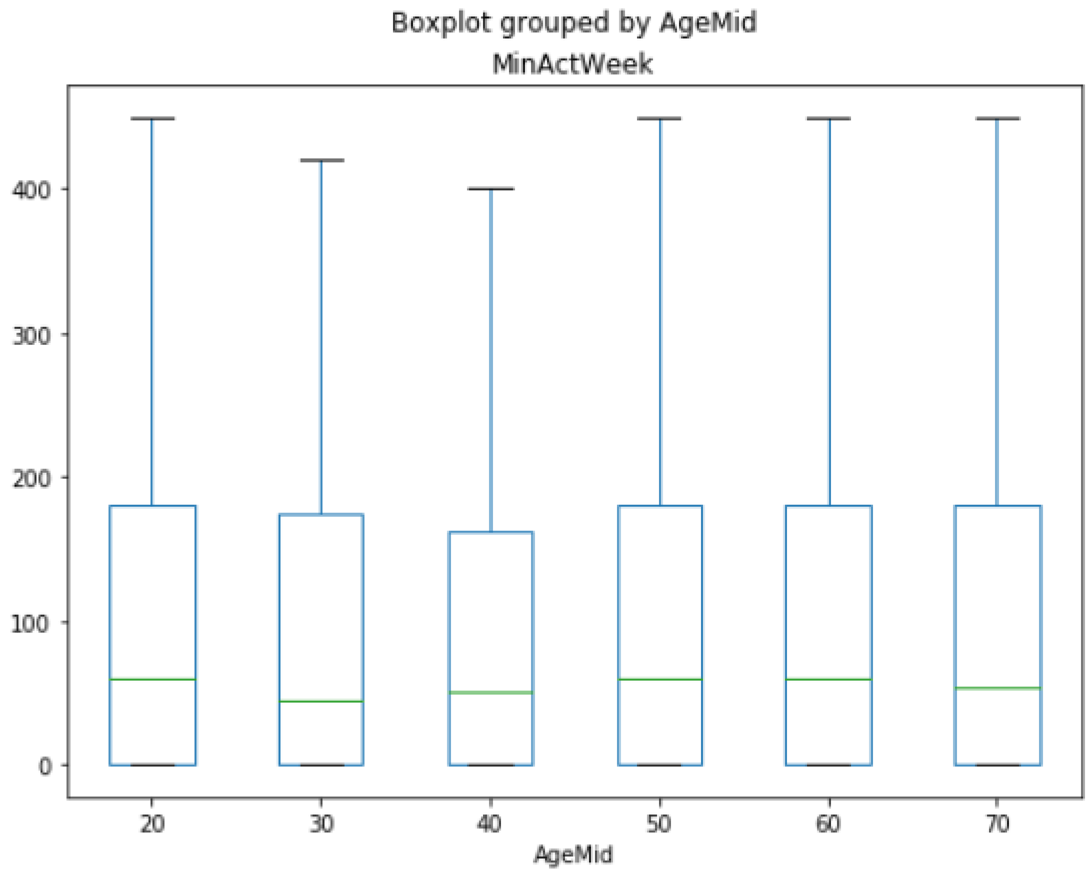
Pandas has various ways to aggregate data, here pd.crosstab() makes a 2 by 2 table of smoking vs sex

Data Visualization (boxplot)

```
In [5]: #boxplot of age bins on X axis, and y is activity per week

        brfss_dat.boxplot(column = 'MinActWeek', by='AgeMid', grid=False, showfliers=False, figsize=(8,6))

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x295c8fa4088>
```



Boxplots show the median (green line), and the inter-quartile range (blue boxes) of the data.

This shows that physical activity is very similar across age groups.

Estimate a Regression Equation

```
In [6]: #Estimating a logistic regression equation

        #Changing sex to dummy variable, regression does not understand text
        brfss_dat['Male'] = 1*(brfss_dat['SEX'] == 'Male')
        ind_vars = ['Male','MinActWeek','AgeMid','CurrentSmoker']

        logit_model = LogisticRegression(penalty='none', solver='newton-cg')
        logit_model.fit(X = brfss_dat[ind_vars], y = brfss_dat['Obese_BMI'])

        print( logit_model.intercept_, logit_model.coef_ )

[-1.28684432] [[-0.06533615 -0.00085874  0.00690985  0.05603047]]
```

The probability of obesity decreases for males and being more active, it increases for older individuals and smokers

$$p(\text{Obese}) = f[-1.3 \ - 0.065(\text{Male}) \ - 0.001(\text{Activity}) + 0.006(\text{Age}) + 0.056(\text{Smoker})]$$

Modelling metrics (Accuracy & Confusion Matrix)

In [7]:

```
#How well do our predictions do
from sklearn.metrics import confusion_matrix

#Getting the predicted probability of obesity per our model
pred_prob = logit_model.predict_proba(X = brfss_dat[ind_vars])[::,1]

#Generating a confusion matrix, setting threshold to predict obese at 30%
con_mat = pd.DataFrame(confusion_matrix(brfss_dat['Obese_BMI'], pred_prob > 0.3),
                       columns=['Predict No','Predict Yes'], index=['Not Obese', 'Obese'])

#The correct guesses are on the diagonal of the confusion matrix
accuracy = (con_mat.iloc[0,0] + con_mat.iloc[1,1] ) / len(brfss_dat)
print("Accuracy")
print("%.2f" % accuracy)

con_mat
```

```
Accuracy
0.69
```

Out[7]:

	Predict No	Predict Yes
Not Obese	7272	961
Obese	2503	420

If we guessed randomly whether people were obese, we would be wrong 50% of the time.

Our model guesses right 69% of the time though.

Apply Predictions to New Data

In [7]:
```
#Apply predictions to newdata

act = range(0,480,60)

new_dat = pd.DataFrame({'Male': 1, 'MinActWeek': act, 'AgeMid': 40, 'CurrentSmoker': 0})
new_dat['PredProbMale'] = logit_model.predict_proba(new_dat)[::,1]
new_dat
```

Out[7]:

	Male	MinActWeek	AgeMid	CurrentSmoker	PredProbMale
0	1	0	40	0	0.254304
1	1	60	40	0	0.244658
2	1	120	40	0	0.235262
3	1	180	40	0	0.226118
4	1	240	40	0	0.217230
5	1	300	40	0	0.208596
6	1	360	40	0	0.200218
7	1	420	40	0	0.192094

The probability of 40 year old non-smoking male with 0 activity per week to be obese is 25%

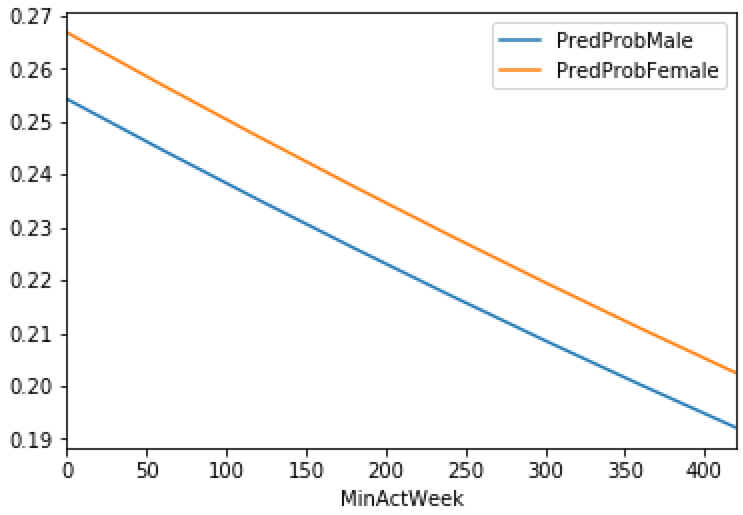
For 420 minutes of activity a week, the probability is only 19%

Model Interpretation

```
In [8]: #Line graph comparing males to females
        new_dat['Male'] = 0
        new_dat['PredProbFemale'] = logit_model.predict_proba(new_dat[ind_vars])[::,1]

        new_dat[['MinActWeek','PredProbMale','PredProbFemale']].plot.line(x='MinActWeek')

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x181838cf648>
```



Males and Females have very similar profiles, males just have a slightly smaller probability of being obese.

Limitations

- We don't evaluate *how well* our predictions do on a new sample, our predictions will be optimistic (will cover in *machine learning 101* how to validate samples)
- Very simple model, some omitted factors (diet), non-linear effects for activity, or interactions among those variables.
- Ignored *missing data* (I threw out missing cases in the dataset for simplicity)
- Weak research design (cross-sectional survey). So should be wary of interpreting as *causal* effects.

Questions?

Future Topics

Have requests?

Let me know!

Introduction to Data Science Course Outline

Andrew Wheeler, PhD, andrew.wheeler@hms.com

- Lesson 01: Data Science 101
- Lesson 02: Machine Learning 101
- Lesson 03: Evaluating Predictions
- Lesson 04: Intro Data Transformation in Python
- Lesson 05: Data Visualization 101
- Lesson 06: Feature Engineering
- Lesson 07: Missing Data
- Lesson 08: Big Data and Parallel Computing Intro
- Lesson 09: Dimension Reduction and Unsupervised Learning
- Lesson 10: High Cardinality (Many Categories)
- Lesson 11: Intro to Forecasting
- Lesson 12: Conducting Experiments

hms®

Data Science 101

Data Science Team

02/27/2020

Andrew Wheeler, PhD

andrew.wheeler@hms.com