Capstone 2 – Final Report

Jan. 8, 2018

1. Problem Definition:

The Palm Beach Code School (PBCS - www.palmbeachcodeschool.com) currently offers a sixteen week Web Developer Program.

There are two main future planning questions to be answered:

a) Can PBCS offer a shorter class sufficient to prepare students for some starting web development positions?
b) And, can they offer a follow-up, more in-depth course for jobs requiring more extensive skills?

The basic strategy is to find key, required job skills for local web developer positions and then use unsupervised, machine learning clustering to determine if they can be divided into two levels of requirements.

2. Client:

The Palm Beach Code School (PBCS - www.palmbeachcodeschool.com).

3. Dataset:

The data base for this project is a python pandas dataframe scraped and then cleaned from Indeed web developer want ads in Palm Beach County, Florida and the surrounding area.

The following data preparation / cleaning steps are required:

a. Read each job page site html into a BeautifulSoup object.
b. Remove all scripts and style elements.
c. Extract details about the job: company, job title, location, pay (rarely available), description url, etc.
d. Use Selenium to simulate picking the job description and load it back into BeautifulSoup. (Note that BeautifulSoup cannot handle real-time modifications to the original DOM description.)
e. Process the key words:
    1. Remove blank lines and ends-of-line.
    2. Remove any Unicode characters.
    3. Remove all non-alphabetic characters. (This has to be modified to handle some skills such as Angular2 or C++.)
    4. Set to lower lase and split them into a list.
    5. Remove stop-words.
    6. Mark the keywords as either 1-True or 0-False if they appear in the description.
f. Simplify / clean the data for input to the clustering algorithm (from 75 -> 46 items):
    1. Remove un-used job skills.
    2. Sort jobs by company / title and remove duplicates (several “sponsored” jobs appear on more than one page).

3. Remove jobs that don't have any of the skills (incomplete job descriptions?).
4. Change boolean to integer values.
5. Remove all columns except job skills.

4. Other potential data sets / projects:

Two additional, adjunct projects could be performed:

a. Conduct another web scrape session at a later date to examine how skill requirements are changing.
b. Compare the skills for local jobs to other areas of the state / country.

5. Results:

75 jobs were processed in the initial web scraping:

```
In [5]: web_site = 'https://www.indeed.com/jobs?q=web+%2
        web_skills_info(web_site, 'West Palm Beach')

        Getting page 1
        Getting page 2
        Getting page 3
        Getting page 4
        Getting page 5
        Done with collecting the job postings!
        There were 75 jobs successfully found.
```

Initial data results sample:

```
df_all = pd.read_csv('WPBWebJobs.csv')
df_all.head(5)
```

	Unnamed: 0	city	job_title	company_name	location	salary	website	Agile	Android	Angular	...	MongoDB	MySQL
0	1	West Palm Beach	Web Programmer Boca Raton, Florida	NCCI Holdings, Inc.	Boca Raton, FL	Not_found	https://www.indeed.com/jobs?q=web+%28developer...	False	False	False	...	False	False
1	2	West Palm Beach	Backend Web Developer	Vazkor Technologies	Boynton Beach, FL 33426	65,000–85,000 a year	https://www.indeed.com/jobs?q=web+%28developer...	False	False	False	...	False	True
2	3	West Palm Beach	Web Designer/Developer	DDG, Inc.	West Palm Beach, FL	Not_found	https://www.indeed.com/jobs?q=web+%28developer...	False	False	False	...	False	False
3	4	West Palm Beach	Web Design / Programmer	George's Music	West Palm Beach, FL	Not_found	https://www.indeed.com/jobs?q=web+%28developer...	False	False	False	...	False	True
4	5	West Palm	Web Programmer Boca Raton, Florida	NCCI Holdings, Inc.	Boca Raton,	Not_found	https://www.indeed.com/jobs?q=web+%28developer...	False	False	False	...	False	False

Cleaning (in Excel) down to just the keywords with integer values for input to machine learning:

```
df_clust = pd.read_csv('JustKeywords.csv')
df_clust.head(5)
```

	Agile	Android	Angular	Bootstrap	C++	Cloud	CSS	Excel	HTML	iOS	...	Mobile	Python	I
0	0	0	0	0	0	0	0	0	0	0	...	0	0	
1	0	0	0	0	0	0	1	0	1	0	...	1	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	0	0	0	1	0	1	0	...	1	0	
4	0	0	0	0	0	0	1	0	1	0	...	1	0	

5 rows × 25 columns

```
df_clust.shape
```

```
(46, 25)
```

Performing KMeans clustering (unsupervised) – note that each cluster contains about 50% of the total (22 in cluster1 – the simpler job requirements and 24 in cluster0 – more advanced):

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
kmeans.fit(df_clust)
y_kmeans = kmeans.predict(df_clust)
```

```
y_kmeans
```

```
array([0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1])
```

```
sum(y_kmeans) # number in cluster 1
```

```
22
```

```
len(y_kmeans) - sum(y_kmeans) # number in cluster 0
```

```
24
```

Transforming the dataframe:

```
df_cl_tots = df_clust.T
df_cl_tots
```

	0	1	2	3	4	5	6	7	8	9	...	36	37	38	39	40	41	42	43	44	45
Agile	0	0	0	0	0	0	0	1	1	0	...	1	0	0	0	1	0	0	0	0	0
Android	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Angular	0	0	0	0	0	1	0	1	1	0	...	0	0	0	0	0	0	0	0	0	0
Bootstrap	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	1	0	0	0	0	0
C++	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
Cloud	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
CSS	0	1	0	1	1	1	0	1	1	1	...	1	1	0	0	1	1	1	0	0	1
Excel	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
HTML	0	1	0	1	1	1	1	1	1	1	...	1	1	1	0	1	1	1	0	1	1

```
# create new column with each cluster's totals per keyword
df_cl_tots['clust0'] = 0
df_cl_tots['clust1'] = 0
for i in range(len(df_cl_tots.index)):
    clu0 = 0
    clu1 = 0
    for j in range(len(df_cl_tots.columns)-2):
        clu0 = clu0 + ((1-y_kmeans[j])*df_cl_tots.iloc[i,j])
        clu1 = clu1 + (y_kmeans[j]*df_cl_tots.iloc[i,j])
    df_cl_tots['clust0'][i] = clu0
    df_cl_tots['clust1'][i] = clu1
df_cl_tots
```

	0	1	2	3	4	5	6	7	8	9	...	38	39	40	41	42	43	44	45	clust0	clust1
Agile	0	0	0	0	0	0	0	1	1	0	...	0	0	1	0	0	0	0	0	5	8
Android	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	2	1
Angular	0	0	0	0	0	1	0	1	1	0	...	0	0	0	0	0	0	0	0	2	3
Bootstrap	0	0	0	0	0	0	0	0	1	0	...	0	0	1	0	0	0	0	0	0	4
C++	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	1
Cloud	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	4	1
CSS	0	1	0	1	1	1	0	1	1	1	...	0	0	1	1	1	0	0	1	3	22
Excel	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	2	0
HTML	0	1	0	1	1	1	1	1	1	1	...	1	0	1	1	1	0	1	1	8	22
iOS	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	1

Final results:

```
# Top ten - clust1
df_cl_tots.sort_values('clust1', ascending=False).head(10)
```

	0	1	2	3	4	5	6	7	8	9	...	38	39	40	41	42	43	44	45	clust0	clust1
CSS	0	1	0	1	1	1	0	1	1	1	...	0	0	1	1	1	0	0	1	3	22
HTML	0	1	0	1	1	1	1	1	1	1	...	1	0	1	1	1	0	1	1	8	22
Javascript	1	1	0	0	1	1	0	1	1	1	...	0	0	1	1	0	1	0	1	6	19
JQuery	1	1	0	0	1	1	0	1	0	1	...	0	0	1	0	0	0	0	0	3	14
Mobile	0	1	0	1	1	1	0	1	0	0	...	1	1	1	0	0	0	0	0	5	13
Agile	0	0	0	0	0	0	0	1	1	0	...	0	0	1	0	0	0	0	0	5	8
PHP	0	1	1	0	1	0	1	1	0	0	...	0	0	0	1	1	1	1	1	12	7
WordPress	0	0	0	0	1	0	0	1	0	0	...	1	0	1	0	0	0	0	0	5	6
Node	0	0	0	0	0	1	0	1	1	0	...	0	0	1	0	0	0	0	0	1	5
Linux	0	1	0	0	0	0	0	0	0	0	...	0	0	1	0	1	0	0	0	3	4

```
# Top ten - clust0
df_cl_tots.sort_values('clust0', ascending=False).head(10)
```

	0	1	2	3	4	5	6	7	8	9	...	38	39	40	41	42	43	44	45	clust0	clust1
SQL	1	0	1	0	0	0	0	0	0	0	...	0	1	0	0	1	1	0	1	13	4
PHP	0	1	1	0	1	0	1	1	0	0	...	0	0	0	1	1	1	1	1	12	7
MySQL	0	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	9	2
HTML	0	1	0	1	1	1	1	1	1	1	...	1	0	1	1	1	0	1	1	8	22
Java	0	0	1	0	0	0	1	0	0	0	...	0	0	0	0	0	0	1	0	7	2
Javascript	1	1	0	0	1	1	0	1	1	1	...	0	0	1	1	0	1	0	1	6	19
Agile	0	0	0	0	0	0	0	1	1	0	...	0	0	1	0	0	0	0	0	5	8
WordPress	0	0	0	0	1	0	0	1	0	0	...	1	0	1	0	0	0	0	0	5	6
Mobile	0	1	0	1	1	1	0	1	0	0	...	1	1	1	0	0	0	0	0	5	13
Cloud	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	4	1

Summary:

Palm Beach Code School should continue offering their current web development course primarily covering CSS, HTML, Javascript, and PHP (perhaps adding Mobile, Agile, and Wordpress) for the entry-level ~50% of web jobs in the Palm Beach County area.

In addition, for the remaining ~50% of higher-level jobs, they should consider offering a more advanced course covering SQL, MySQL, Java, advanced PHP, and Cloud based skills.

Appendix A: Weka clustering results:

During data exploration, clustering was also performed on Weka with very similar results.

=== Run information ===

Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100

-periodic-pruning 10000 -min-density 2.0 -t1 -1.25

-t2 -1.0 -N 2 -A "weka.core.EuclideanDistance

-R first-last" -I 500 -num-slots 1 -S 10

Relation: WPBWebJobsBinaryToInt

Instances: 46

Attributes: 26- Agile, Android, Angular, Bootstrap, C++, Cloud,

CSS, Excel, HTML, iOS, Java, Javascript, JQuery,

JSON, Linux, Mobile, Python, MongoDB, MySQL,

Node, PHP, SQL, Windows, WordPress, XML

Ignored: JID

Test mode: evaluate on training data

=== Clustering model (full training set) ===

kMeans

======

Number of iterations: 3

Within cluster sum of squared errors: 128.89278752436644

Initial starting points (random):

Cluster 0: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0

Cluster 1: 0,0,0,0,0,0,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0

Missing values globally replaced with mean/mode

Final cluster centroids:

	Cluster#		
Attribute	Full Data	0	1
	(46.0)	(19.0)	(27.0)
Agile	0.2826	0.1579	0.3704
Android	0.0652	0.0526	0.0741
Angular	0.1087	0.1053	0.1111
Bootstrap	0.087	0	0.1481
C++	0.0217	0	0.037

Cloud	0.1087	0.2105	0.037
CSS	0.5435	0	0.9259
Excel	0.0435	0.0526	0.037
HTML	0.6522	0.1579	1
iOS	0.0435	0	0.0741
Java	0.1957	0.2632	0.1481
Javascript	0.5435	0.2632	0.7407
JQuery	0.3696	0.0526	0.5926
JSON	0.087	0	0.1481
Linux	0.1522	0.0526	0.2222
Mobile	0.3913	0.2632	0.4815
Python	0.087	0.1053	0.0741
MongoDB	0.0435	0.0526	0.037
MySQL	0.2391	0.2632	0.2222
Node	0.1304	0.0526	0.1852
PHP	0.413	0.4211	0.4074
SQL	0.3696	0.4737	0.2963
Windows	0.0435	0	0.0741
WordPress	0.2391	0.2632	0.2222
XML	0.0652	0	0.1111

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0	19	(41%)
1	27	(59%)