MCIS6273 Data Mining (Prof. Maull) / Fall 2017 / HW1 SOLUTION

This assignment is worth up to 20 POINTS to your grade total if you complete it on time.

Points Possible	Due Date	Time Commitment (estimated)
20	Saturday, Sep 30 @ Midnight	*up to* 10 hours

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by Univerisity or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

OBJECTIVES

- Work with core Pandas and Scikit-Learn concepts in data, data types, representation of data and plotting data
- Explore concepts in statistical inference over real data within Scikit-Learn
- Work with data to understand distance metrics in Scikit-Learn and the impact various metrics have on the outcomes

WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, clone the course repository and modify the `hw1.ipynb` file in the `homework/hw1` directory. If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using Github and cloning repositories.

Turn in a copy of a `.ipynb` file, a PDF or Word Document to Blackboard with the answers to the questions labeled with the § sign.

ASSIGNMENT TASKS

(20%) Work with core Pandas and Scikit-Learn concepts in data, data types, representation of data and plotting data

A great deal of time doing data mining involves understanding the data in a dataset and preprocessing it in preparation for working with it in a real analysis of some sort. We will develop an understanding of:

- how to empirically perform and understand the descriptive statistics of a dataset,
- understand how to reason about data features,
- understand how to use distance metrics,
- understand binarizaration contexts abd techniques, and
- understand how to randomly sample datasets to understand the distribution of data.

We'll be working a new much smaller dataset for this task. Please check the repository for the `bank.csv` data set. There are fewer than 1000 rows to work with in this data. You will need to use Pandas and Scikit-Learn for all the questions here, and it is recommended you take a look at the optional cheat sheets for lecture 2 in the syllabus:

- Pandas Cookbook Github Repo
- Pandas Cheatsheet @ DataCamp.com

§ In the readings and lecture, we talked about distributions of data. Please produce the frequency distribution (histogram) for the following: (1) rural income, (2) car ownership. You will most certainly need to use the `pandas.Series.plot.hist()` method.

§ Produce the scatter plot for income vs age – it doesn't matter which is on the x and y axis. You will benefit from the `DataFrame.plot.scatter()` method.

(40%) Explore concepts in statistical inference over real data within Scikit-Learn

For questions 1-3: Just answer the question over the entire data as requested.

For questions 4 and 5: Consider this scenario …

> The bank is looking for candidates who might be good for sending mortgage loan information to – that is they are looking for people who are **not homeowners** (i.e. don't own a home), but who have the income and other characteristics of potentially good borrowers.
>
> Your task is to look at all the data for people who have no mortgage and build the case for the profile of the borrower (from this dataset) that they should start calling, sending mail and advertising to.
>
> Imagine you are a real-world analyst and will need to do the following:

- build the dataset that contains non-homeowners (i.e. no mortgage)
- report on the characteristics of non-homeowners based on what is being asked

```
import pandas as pd
df = pd.read_csv("./data/bank-data.csv")
```

§ What is the median age of a `UNMARRIED`, `FEMALE`, `HOME OWNER` in the `SUBURBAN` region?

```
df.head()
```

```
<tr style="text-align: right;">
  <th></th>
  <th>id</th>
  <th>age</th>
  <th>sex</th>
  <th>region</th>
  <th>income</th>
  <th>married</th>
  <th>children</th>
  <th>car</th>
  <th>save_act</th>
  <th>current_act</th>
  <th>mortgage</th>
  <th>pep</th>
</tr>
```

```
<tr>
  <th>0</th>
  <td>ID12101</td>
  <td>48</td>
  <td>FEMALE</td>
  <td>INNER_CITY</td>
  <td>17546.0</td>
```

```
    <td>NO</td>
    <td>1</td>
    <td>NO</td>
    <td>NO</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>1</th>
    <td>ID12102</td>
    <td>40</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>30085.1</td>
    <td>YES</td>
    <td>3</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>2</th>
    <td>ID12103</td>
    <td>51</td>
    <td>FEMALE</td>
    <td>INNER_CITY</td>
    <td>16575.4</td>
    <td>YES</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>3</th>
    <td>ID12104</td>
    <td>23</td>
    <td>FEMALE</td>
    <td>TOWN</td>
    <td>20375.4</td>
    <td>YES</td>
    <td>3</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
```

```
  <th>4</th>
  <td>ID12105</td>
  <td>57</td>
  <td>FEMALE</td>
  <td>RURAL</td>
  <td>50576.3</td>
  <td>YES</td>
  <td>0</td>
  <td>NO</td>
  <td>YES</td>
  <td>NO</td>
  <td>NO</td>
  <td>NO</td>
</tr>
```

```
df[(df.married=='NO') & (df.sex=='FEMALE') & (df.mortgage=='YES') & (df.region=='SUBURBAN')].age.descri
```

```
52.0
```

§ Given this bank datset what is the joint probability $\Pr(\text{sex} = MALE \wedge \text{income} \geq 50,000)$? How does this compare with $\Pr(\text{sex} = FEMALE \wedge \text{income} \geq 50,000)$?

Recall:

$$\Pr(A \wedge B) = \Pr(A \big| B) \Pr(B)$$

or

$$\Pr(A \wedge B) = \Pr(A) \Pr(B)$$

, when A and B are independent.

```
pr_male_incgt50k = \
    df[(df.sex=='MALE')&(df.income>=50000)].shape[0] / df[df.income>=50000].shape[0]
```

```
pr_incgt50k = df[df.income>=50000].shape[0] / df.shape[0]
```

```
pr_joint = pr_male_incgt50k * pr_incgt50k
pr_joint
```

```
0.03
```

§ What is conditional probability $\Pr(car = \text{TRUE} \big| sex = \text{MALE} \wedge income \geq 50,000)$?

Since we have already computed the condition $sex = MALE \wedge income \geq 50000$, we need only find the number of these which have cars:

```
df[(df.car=='YES') & (df.sex=='MALE') & (df.income>=50000)]
```

```
<tr style="text-align: right;">
  <th></th>
  <th>id</th>
  <th>age</th>
  <th>sex</th>
  <th>region</th>
  <th>income</th>
  <th>married</th>
  <th>children</th>
  <th>car</th>
  <th>save_act</th>
  <th>current_act</th>
```

```
    <th>mortgage</th>
    <th>pep</th>
  </tr>
  <tr>
    <th>42</th>
    <td>ID12143</td>
    <td>67</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>55716.5</td>
    <td>NO</td>
    <td>2</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>206</th>
    <td>ID12307</td>
    <td>63</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>59409.1</td>
    <td>NO</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>226</th>
    <td>ID12327</td>
    <td>67</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>50474.6</td>
    <td>YES</td>
    <td>2</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>236</th>
    <td>ID12337</td>
    <td>65</td>
    <td>MALE</td>
```

```
    <td>TOWN</td>
    <td>52255.9</td>
    <td>NO</td>
    <td>2</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>282</th>
    <td>ID12383</td>
    <td>64</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>54314.5</td>
    <td>YES</td>
    <td>1</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>306</th>
    <td>ID12407</td>
    <td>63</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>52117.3</td>
    <td>NO</td>
    <td>2</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>342</th>
    <td>ID12443</td>
    <td>64</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>53104.3</td>
    <td>YES</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
```

```
</tr>
<tr>
  <th>580</th>
  <td>ID12681</td>
  <td>63</td>
  <td>MALE</td>
  <td>SUBURBAN</td>
  <td>51879.3</td>
  <td>YES</td>
  <td>2</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>YES</td>
  <td>YES</td>
</tr>
```

```
df[(df.car=='YES') & (df.sex=='MALE') & (df.income>=50000)].count()[0] / \
    df[(df.sex=='MALE') & (df.income>=50000)].count()[0]
```

```
0.44444444444444442
```

§ If the lending requirements were a savings account and income greater than $45,000, what about this data might make it difficult to justify any campaign at all? (**HINT:** most mortgages are 30-years in duration) Provide **concrete evidence** for your claims; you may do this in a variety of ways *including describing the statistic that leads you to your claim.*

```
df[(df.save_act=='YES') & (df.income>=45000)].describe()
```

```
<tr style="text-align: right;">
  <th></th>
  <th>age</th>
  <th>income</th>
  <th>children</th>
</tr>

<tr>
  <th>count</th>
  <td>74.000000</td>
  <td>74.000000</td>
  <td>74.000000</td>
</tr>
<tr>
  <th>mean</th>
  <td>61.189189</td>
  <td>52170.395946</td>
  <td>1.135135</td>
</tr>
<tr>
  <th>std</th>
  <td>4.898073</td>
  <td>4678.395935</td>
  <td>1.051083</td>
</tr>
<tr>
  <th>min</th>
  <td>47.000000</td>
```

```
    <td>45031.900000</td>
    <td>0.000000</td>
  </tr>
  <tr>
    <th>25%</th>
    <td>58.000000</td>
    <td>48732.850000</td>
    <td>0.000000</td>
  </tr>
  <tr>
    <th>50%</th>
    <td>63.000000</td>
    <td>51291.800000</td>
    <td>1.000000</td>
  </tr>
  <tr>
    <th>75%</th>
    <td>64.000000</td>
    <td>56263.000000</td>
    <td>2.000000</td>
  </tr>
  <tr>
    <th>max</th>
    <td>67.000000</td>
    <td>63130.100000</td>
    <td>3.000000</td>
  </tr>
```

There are 74 people in this group. The median age however is 61.19! The min is 47! Even if you consider the youngest of this group, they would be in their late 70s (47 + 30-year mortgage), making them well into retirement by the time the home was even paid off! Thus, this is not a good group to consider.

§ If the savings account requirement was ignored and the income lowered to a minimum of $30,000, to whom (`MALE` or `FEMALE`) and in which region (`RURAL`, `TOWN`, `INNER_CITY`, `SUBURBAN`) would the bank likely have the best success?

There are several ways to look at this. First, we'll just find the people under 47 and sort by region and sex, then take a look at their statistics.

```
df[(df.income>=30000)&(df.mortgage=='NO')&(df.age<47)].sort_values(by=['sex','region'])
```

```
<tr style="text-align: right;">
    <th></th>
    <th>id</th>
    <th>age</th>
    <th>sex</th>
    <th>region</th>
    <th>income</th>
    <th>married</th>
    <th>children</th>
    <th>car</th>
    <th>save_act</th>
    <th>current_act</th>
    <th>mortgage</th>
    <th>pep</th>
  </tr>
```

```
<tr>
  <th>222</th>
  <td>ID12323</td>
  <td>42</td>
  <td>FEMALE</td>
  <td>INNER_CITY</td>
  <td>39308.7</td>
  <td>YES</td>
  <td>1</td>
  <td>NO</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>YES</td>
</tr>
<tr>
  <th>291</th>
  <td>ID12392</td>
  <td>44</td>
  <td>FEMALE</td>
  <td>INNER_CITY</td>
  <td>39253.6</td>
  <td>NO</td>
  <td>0</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>NO</td>
  <td>YES</td>
</tr>
<tr>
  <th>371</th>
  <td>ID12472</td>
  <td>44</td>
  <td>FEMALE</td>
  <td>INNER_CITY</td>
  <td>38453.7</td>
  <td>NO</td>
  <td>2</td>
  <td>NO</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>YES</td>
</tr>
<tr>
  <th>381</th>
  <td>ID12482</td>
  <td>46</td>
  <td>FEMALE</td>
  <td>INNER_CITY</td>
  <td>32583.5</td>
  <td>YES</td>
  <td>2</td>
```

```
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>538</th>
    <td>ID12639</td>
    <td>43</td>
    <td>FEMALE</td>
    <td>INNER_CITY</td>
    <td>38784.0</td>
    <td>YES</td>
    <td>0</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>123</th>
    <td>ID12224</td>
    <td>35</td>
    <td>FEMALE</td>
    <td>RURAL</td>
    <td>33028.3</td>
    <td>NO</td>
    <td>1</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>425</th>
    <td>ID12526</td>
    <td>44</td>
    <td>FEMALE</td>
    <td>RURAL</td>
    <td>30488.7</td>
    <td>YES</td>
    <td>0</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>442</th>
    <td>ID12543</td>
```

```
    <td>43</td>
    <td>FEMALE</td>
    <td>RURAL</td>
    <td>36281.0</td>
    <td>YES</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>449</th>
    <td>ID12550</td>
    <td>38</td>
    <td>FEMALE</td>
    <td>RURAL</td>
    <td>34061.4</td>
    <td>NO</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>558</th>
    <td>ID12659</td>
    <td>38</td>
    <td>FEMALE</td>
    <td>SUBURBAN</td>
    <td>31290.6</td>
    <td>YES</td>
    <td>0</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>233</th>
    <td>ID12334</td>
    <td>32</td>
    <td>FEMALE</td>
    <td>TOWN</td>
    <td>30404.3</td>
    <td>YES</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
```

```
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>383</th>
    <td>ID12484</td>
    <td>40</td>
    <td>FEMALE</td>
    <td>TOWN</td>
    <td>34836.8</td>
    <td>YES</td>
    <td>1</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>447</th>
    <td>ID12548</td>
    <td>35</td>
    <td>FEMALE</td>
    <td>TOWN</td>
    <td>30799.5</td>
    <td>YES</td>
    <td>2</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>587</th>
    <td>ID12688</td>
    <td>43</td>
    <td>FEMALE</td>
    <td>TOWN</td>
    <td>31273.8</td>
    <td>NO</td>
    <td>1</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>85</th>
    <td>ID12186</td>
    <td>36</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
```

```
    <td>31683.1</td>
    <td>YES</td>
    <td>1</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>514</th>
    <td>ID12615</td>
    <td>37</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>33886.4</td>
    <td>NO</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>552</th>
    <td>ID12653</td>
    <td>45</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>41107.2</td>
    <td>YES</td>
    <td>2</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>571</th>
    <td>ID12672</td>
    <td>40</td>
    <td>MALE</td>
    <td>INNER_CITY</td>
    <td>36166.2</td>
    <td>YES</td>
    <td>0</td>
    <td>NO</td>
    <td>NO</td>
    <td>NO</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
```

```
<tr>
  <th>445</th>
  <td>ID12546</td>
  <td>40</td>
  <td>MALE</td>
  <td>RURAL</td>
  <td>31864.8</td>
  <td>YES</td>
  <td>0</td>
  <td>YES</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>NO</td>
</tr>
<tr>
  <th>541</th>
  <td>ID12642</td>
  <td>39</td>
  <td>MALE</td>
  <td>RURAL</td>
  <td>37389.0</td>
  <td>YES</td>
  <td>2</td>
  <td>NO</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>YES</td>
</tr>
<tr>
  <th>34</th>
  <td>ID12135</td>
  <td>43</td>
  <td>MALE</td>
  <td>SUBURBAN</td>
  <td>37521.9</td>
  <td>NO</td>
  <td>0</td>
  <td>NO</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>YES</td>
</tr>
<tr>
  <th>105</th>
  <td>ID12206</td>
  <td>43</td>
  <td>MALE</td>
  <td>TOWN</td>
  <td>32395.5</td>
  <td>YES</td>
  <td>3</td>
```

```
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>110</th>
    <td>ID12211</td>
    <td>36</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>37330.5</td>
    <td>NO</td>
    <td>2</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>113</th>
    <td>ID12214</td>
    <td>36</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>33630.6</td>
    <td>NO</td>
    <td>2</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>151</th>
    <td>ID12252</td>
    <td>43</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>36432.8</td>
    <td>NO</td>
    <td>2</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>176</th>
    <td>ID12277</td>
```

```
    <td>40</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>37558.5</td>
    <td>YES</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>218</th>
    <td>ID12319</td>
    <td>39</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>31693.5</td>
    <td>NO</td>
    <td>0</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
    <td>YES</td>
  </tr>
  <tr>
    <th>310</th>
    <td>ID12411</td>
    <td>46</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>30658.7</td>
    <td>YES</td>
    <td>0</td>
    <td>NO</td>
    <td>YES</td>
    <td>YES</td>
    <td>NO</td>
    <td>NO</td>
  </tr>
  <tr>
    <th>479</th>
    <td>ID12580</td>
    <td>46</td>
    <td>MALE</td>
    <td>TOWN</td>
    <td>43395.5</td>
    <td>NO</td>
    <td>1</td>
    <td>YES</td>
    <td>YES</td>
    <td>YES</td>
```

```
  <td>NO</td>
  <td>YES</td>
</tr>
```

The largest, youngest group are `MALES` in `TOWN`. They have the income and the risk profile (e.g. savings account). However the median age is an issue:

```python
df[(df.income>=30000)&(df.mortgage=='NO')&(df.age<47)&(df.sex=='MALE')&(df.region=='TOWN')].describe()[
```

```
41.5
```

They'd be in their 70s by the time the mortgage was paid off. Perhaps we should try to see what's going on with those just a but younger ...

We could also go to the below 35 age since many people ideally would like to have their home paid off before US retirement age.

```python
df[(df.income>=30000)&(df.mortgage=='NO')&(df.age<35)].sort_values(by=['sex','region'])
```

```
<tr style="text-align: right;">
  <th></th>
  <th>id</th>
  <th>age</th>
  <th>sex</th>
  <th>region</th>
  <th>income</th>
  <th>married</th>
  <th>children</th>
  <th>car</th>
  <th>save_act</th>
  <th>current_act</th>
  <th>mortgage</th>
  <th>pep</th>
</tr>
```

```
<tr>
  <th>233</th>
  <td>ID12334</td>
  <td>32</td>
  <td>FEMALE</td>
  <td>TOWN</td>
  <td>30404.3</td>
  <td>YES</td>
  <td>0</td>
  <td>YES</td>
  <td>YES</td>
  <td>YES</td>
  <td>NO</td>
  <td>NO</td>
</tr>
```

In this group we only have `FEMALE` in `TOWN` but only one, so a campaign may not be worth the time.

(40%) Work with data to understand distance metrics in Scikit-Learn and the impact various metrics have on the outcomes

A common thing to understand in a dataset is to determine some number of *nearest neighbors* to a data point. We will see this come back when we get to clustering. For now, let's explore the NearestNeighbor implementation in Scikit-Learn.

You will most certainly need to convert the categorical non-numeric data to binary features, and you may be served well by reading about preprocessing data in Scikit-Learn and also taking a closer look at `sklearn.preprocessing.LabelBinarizer` and `sklearn.preprocessing.OneHotEncoder`. You might also find this resource of value if you read only the first few slides on binarization.

§ What are the 5 nearest neighbors (the indices) to index #7, #21, #40 and #94? Your output will be a list of the tuples of the 5 closest indices and their values (e.g. `[(5, 4.56356), (19, 8.83452), (233, 12.23486), ...]`). Use the *minkowski (default) distance* first.

§ Use the *cosine similarity metric* and *euclidean distance metric* (e.g. invoke `NearestNeighbor(..., metric='cosine')` and `NearestNeighbor(..., metric='euclidean')`). Produce the same lists for the same indices as in #1. What are the differences in the nearest neighbor lists? Which seem to be the most similar? Provide your thoughts on why there are differences?

```python
df = df.drop('id', 1)
df_binarized = pd.get_dummies(df)


def compute_nn_report(nn_list=[7, 21, 40, 90], metric=None, k=5):
    from sklearn.neighbors import NearestNeighbors

    if metric:
        nn = NearestNeighbors(k, metric=metric, algorithm='brute')
    else:
        nn = NearestNeighbors(k)

    nn_matrix = nn.fit(df_binarized).kneighbors(df_binarized)

    for i in nn_list:
        print("The nearest neighbors to {} are: {} \n\tevidence: {}" \
              .format(i, nn_matrix[1][i], tuple(zip(nn_matrix[1][i], nn_matrix[0][i]))))
```

```python
compute_nn_report(metric='minkowski')
```

```
The nearest neighbors to 7 are: [  7  56 164 227 565]
    evidence: ((7, 0.0), (56, 31.09356846588663), (164, 46.354719284992584), (227, 58.485553771361175),
The nearest neighbors to 21 are: [ 21 436 413   2 483]
    evidence: ((21, 0.0), (436, 23.618636709881851), (413, 32.306191357038912), (2, 78.170390813623925)
The nearest neighbors to 40 are: [ 40 100 168 335 599]
    evidence: ((40, 0.0), (100, 66.345233437964566), (168, 69.172899317476165), (335, 86.82862431202590
The nearest neighbors to 90 are: [ 90 377  60 559 565]
    evidence: ((90, 0.0), (377, 10.04987562112089), (60, 51.521257748020972), (559, 53.23729895401285),
```

```python
compute_nn_report(metric='euclidean')
```

```
The nearest neighbors to 7 are: [  7  56 164 227 565]
    evidence: ((7, 0.0), (56, 31.09356846588663), (164, 46.354719284992584), (227, 58.485553771361175),
The nearest neighbors to 21 are: [ 21 436 413   2 483]
    evidence: ((21, 0.0), (436, 23.618636709881851), (413, 32.306191357038912), (2, 78.170390813623925)
The nearest neighbors to 40 are: [ 40 100 168 335 599]
    evidence: ((40, 0.0), (100, 66.345233437964566), (168, 69.172899317476165), (335, 86.82862431202590
The nearest neighbors to 90 are: [ 90 377  60 559 565]
    evidence: ((90, 0.0), (377, 10.04987562112089), (60, 51.521257748020972), (559, 53.23729895401285),
```

```python
compute_nn_report(metric='cosine')
```

```
The nearest neighbors to 7 are: [  7  38 592 524 227]
```

```
    evidence: ((7, 2.2204460492503131e-16), (38, 2.9460347494847383e-09), (592, 3.2576520370142248e-09)
The nearest neighbors to 21 are: [ 21 202 128 132 314]
    evidence: ((21, 0.0), (202, 1.5016694221436921e-08), (128, 1.6308320693880773e-08), (132, 1.7706350
The nearest neighbors to 40 are: [ 40 533 100 141 387]
    evidence: ((40, 0.0), (533, 3.1807753098078706e-09), (100, 4.020875410404301e-09), (141, 6.75708522
The nearest neighbors to 90 are: [ 90 185 194 517 599]
    evidence: ((90, 2.2204460492503131e-16), (185, 2.9281650437695816e-09), (194, 3.3899721918473347e-0
```