UDACITY

Back to Machine Learning Engineer Nanodegree

Creating Customer Segments

REVIEW

HISTORY

Meets Specifications

Hi there, it's Cláudio! Thanks for sending all the required files for the review process and for all code executing without any problem.

Congratulations for your project submission and for the quality presented in this challenge. You really did a great job.

I hope you had enjoyed doing this project and put in practice important concepts from machine learning. I will leave my contact below in case you have any doubt about this review as well to stay connected.

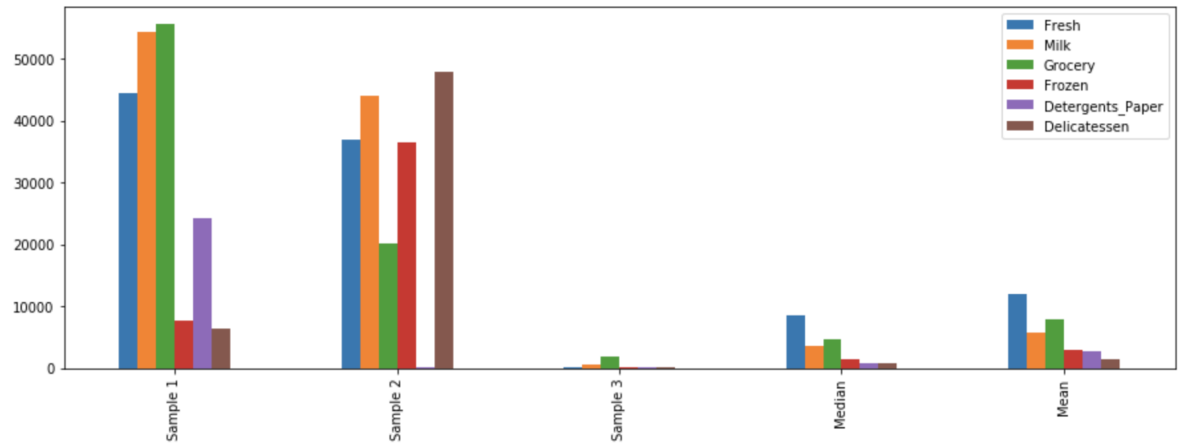
That's all. Enjoy machine learning and keep it up the great work. I will look forward for your next project submission.

Finally, I wanted to share a interesting tool from google that helps machine learning engineers to understand the data really fast and then make a decision on what type of algorithm it will fit better that data, called: Facets - Visualizations (https://pair-code.github.io/facets/). Definitely check it out.

The power of machine learning comes from its ability to learn patterns from large amounts of data. Understanding your data is critical to building a powerful machine learning system.

Facets contains two robust visualizations to aid in understanding and analyzing machine learning datasets. Get a sense of the shape of each feature of your dataset using Facets Overview, or explore individual observations using Facets Dive.

```
labels = ['Sample 1', 'Sample 2', 'Sample 3', 'Median', 'Mean']
samples_for_plot.plot(kind='bar', figsize=(15, 5))
plt.xticks(range(5), labels)
plt.show()
```



A prediction score for the removed feature is accurately reported. Justification is made for whether the removed feature is relevant.

Good job:

> Grocery and Detergents_Paper have some positive score (approximately 0.60). Milk has almost no score (approximately 0.10). The other three are not predictable (negative scores).

Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

Great job.

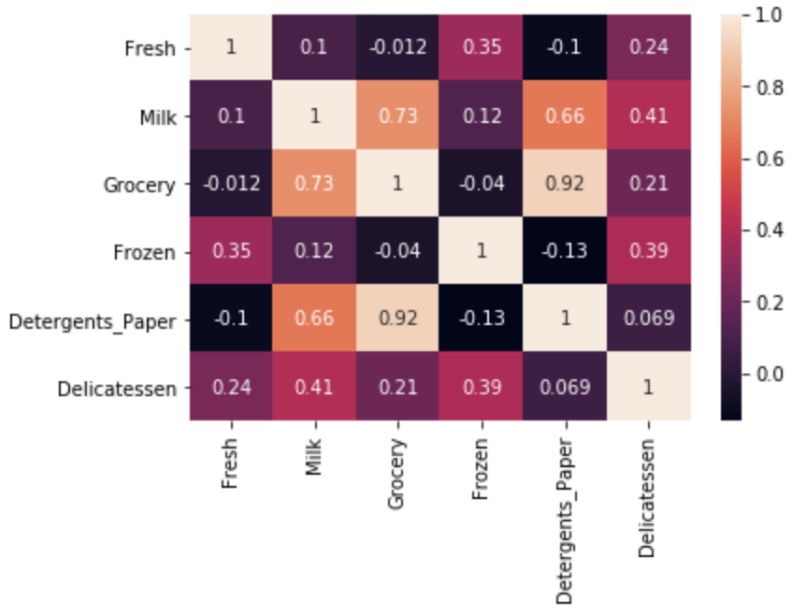
> Grocery <-> Detergents_Paper are correlated. Milk <-> Detergents_Paper, and Milk <-> Grocery are also correlated but not to the same degree.

Suggestion:

- I would recommend you to implement the heat map with values which makes this job easier, take a look into an example below:

In [6]:
```
#Adding graphs for correlation to help out throught the process
import seaborn as sns
sns.heatmap(data.corr(), annot=True)
```

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1a732790>



Data Preprocessing

Feature scaling for both the data and the sample data has been properly implemented in code.

Correctly implemented as expected:

```
# TODO: Scale the data using the natural logarithm
log_data = np.log(data)

# TODO: Scale the sample data using the natural logarithm
log_samples = np.log(samples)
```

Bonus:

- Here I will leave some good articles about the importance of feature scaling:
  http://sebastianraschka.com/Articles/2014_about_feature_scaling.html
  https://stackoverflow.com/questions/26225344/why-feature-scaling
  https://www.quora.com/Why-do-we-use-standardization-for-feature-scaling-for-machine-learning-algorithms

Student identifies extreme outliers and discusses whether the outliers should be removed. Justification is made for any data points removed.

Great job. You have correctly identified the outliers and removed them all:

> The indices of the five data points which are double-counted are [65, 66, 75, 128, 154].

Bonus:

- Here I will leave a great article about to drop or not outliers, definitely check it out: http://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/

Feature Transformation

The total variance explained for two and four dimensions of the data from PCA is accurately reported. The first four dimensions are interpreted as a representation of customer spending with justification.

Good job. You have correctly identified the total variance for cumulative. You have find out this precisely:

> The cumulative explained variance for two and four dimensions is approximately 71% and 93%, respectively. This can change by a few percent based on what outliers are removed.

Bonus:

- Here I will leave good articles about PCA analysis:
  https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32
  https://stats.stackexchange.com/questions/132976/does-pca-mean-selecting-most-important-features-and-ignoring-the-others
  http://abhijitannaldas.com/dimensionality-reduction-and-principal-component-analysis-pca-explained.html

PCA has been properly implemented and applied to both the scaled data and scaled sample data for the two-dimensional case in code.

Good implementing this correctly:

```
# TODO: Apply PCA by fitting the good data with only two dimensions
pca = PCA(n_components=2)
pca.fit(good_data)

# TODO: Transform the good data using the PCA fit above
reduced_data = pca.transform(good_data)
```

```
# TODO: Transform log_samples using the PCA fit above
pca_samples = pca.transform(log_samples)
```

Clustering

The Gaussian Mixture Model and K-Means algorithms have been compared in detail. Student's choice of algorithm is justified based on the characteristics of the algorithm and data.

Good answer and concise one. Good choice on GMM.

Suggestion:

- It's always a good idea to link references, images an business cases in order to convey a clearer message to all type of audience dealing with the problem and it's solution. Try to elaborate more on these type of discussions.
- Regarding your choice of algorithm:
  Both the algorithms will do fine here, although considering the fact that there are no visually separable clusters in the biplot, one might, indeed, prefer the soft-clustering approach of GMM, particularly since the dataset is quite small and scalability is not an issue.
- For large datasets, an alternative strategy could be to go with the faster KMeans for preliminary analysis, and if you later think that the results could be significantly improved, use GMM in the next step while using the cluster assignments and centres obtained from KMeans as the initialisation for GMM. In fact, many implementations of GMM automatically perform this preliminary step for initialisation.
- I provide below some citations which might prove useful, if you would like to go deeper into the dynamics of these algorithms:
  http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/mixture.html
  http://www.nickgillian.com/wiki/pmwiki.php/GRT/GMMClassifier
  http://playwidtech.blogspot.hk/2013/02/k-means-clustering-advantages-and.html
  http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/K-Means_Clustering_Overview.htm
  http://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means
  http://www.r-bloggers.com/k-means-clustering-is-not-a-free-lunch/
  http://www.r-bloggers.com/pca-and-k-means-clustering-of-delta-aircraft/
  https://shapeofdata.wordpress.com/2013/07/30/k-means/
  http://mlg.eng.cam.ac.uk/tutorials/06/cb.pdf

Several silhouette scores are accurately reported, and the optimal number of clusters is chosen based on the best reported score. The cluster visualization provided produces the optimal number of clusters based on the clustering algorithm chosen.

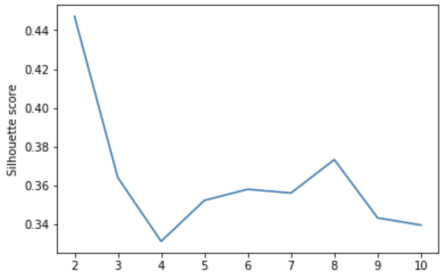
Great job. You have identified the best number of silhouettes properly.

Two clusters will almost always give the best Silhouette score of approximately 0.42 (depending on what outliers were removed).

Suggestion:

- Would be a great idea to plot in some visual the number versus score for a best management for the decision taken, for example:

In [19]:
```
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score

# TODO: Aplique o algoritmo de clustering de sua escolha aos dados reduzidos
clusters_scores = []
for clusters in np.arange(2,11):

    clusterer = KMeans(n_clusters=clusters).fit(reduced_data)

    # TODO: Preveja o cluster para cada ponto de dado
    preds = clusterer.predict(reduced_data)

    # TODO: Ache os centros do cluster
    centers = clusterer.cluster_centers_

    # TODO: Preveja o cluster para cada amostra de pontos de dado transformados
    sample_preds = clusterer.predict(pca_samples)

    # TODO: Calcule a média do coeficiente de silhueta para o número de clusters escolhidos
    score = silhouette_score(reduced_data, preds)

    #Appending to the list
    clusters_scores.append(score)

    #Setting for the best score
    best_cluster = clusters_scores.index(max(clusters_scores)) + 2
    clusterer = KMeans(n_clusters=best_cluster).fit(reduced_data)
    preds = clusterer.predict(reduced_data)
    sample_preds = clusterer.predict(pca_samples)
    centers = clusterer.cluster_centers_

#ploting the graph of silhouette scores

plt.plot(np.arange(2,11),clusters_scores)
plt.xlabel("Quantity of clusters")
plt.ylabel("Silhouette score")
plt.show()

print("The best cluster number is {} with a silhoute score of {}".format(best_cluster, max(clusters_scores)))
```



The establishments represented by each customer segment are proposed based on the statistical description of the dataset. The inverse transformation and inverse scaling has been properly implemented and applied to the cluster centers in code.

Very well implemented:

```
# TODO: Inverse transform the centers
log_centers = pca.inverse_transform(centers)

# TODO: Exponentiate the centers
true_centers = np.exp(log_centers)
```

Sample points are correctly identified by customer segment, and the predicted cluster for each sample point is discussed.

One interesting point to note from the cluster_visualization plot is that the two clusters are essentially separated by a value on the first PCA dimension, which we saw earlier is predominantly a combination of Detergents_Paper, Grocery and Milk. The rest of the features, which figure prominently only in the second PCA dimension, don't really matter!

Conclusion

Student correctly identifies how an A/B test can be performed on customers after a change in the wholesale distributor's service.

Great explanation.

Bonus:

- Here I will leave some articles about A/B testing in AI and how their are complementing each other:
  https://hackernoon.com/ai-as-complement-to-a-b-test-design-e8f4b5e28d92
  https://www.dynamicyield.com/ab-testing/
  https://www.mediapost.com/publications/article/305837/ab-testing-vs-ai-conversions.html

Student discusses with justification how the clustering data can be used in a supervised learner for new predictions.

Good job. Your answer is accurate.

> The 'customer segment' labels can be used as an additional input feature, which a supervised learner could train on and then make predictions for the new customers.

Comparison is made between customer segments and customer 'Channel' data. Discussion of customer segments being identified by 'Channel' data is provided, including whether this representation is consistent with previous results.

Good job. Answering to your question it might be too. Check it out these great discussion about it:

https://stats.stackexchange.com/questions/316199/pca-and-visualization-using-biplots-on-data-with-mixed-types
https://stackoverflow.com/questions/16705229/reversing-the-axis-range-in-3d-graph-in-python
https://sukhbinder.wordpress.com/2015/08/05/biplot-with-python/

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review