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.

Ask:
Are you going to evaluate this review? This is very important to me and Udacity. We take so seriously the student's experience that we are continuing learning from you.
Not going to give a 5 stars? Please let me know what opportunities to improve I can make for the next reviews. Let me know what you didn't like so I can always improve.
5 stars? WOW! Super thanks! That would be great to hear from you. Please, also let me know what you liked the most so I can keep it.

Email: cgimenest@uol.com.br
Linkedin: https://www.linkedin.com/in/claudiogimenestoledo/

Data Exploration

Three separate samples of the data are chosen and their establishment representations are proposed based on the statistical description of the dataset.

Great job selecting samples of the dataset:

> indices = [0, 100, 200]

This is an important step to consider during the exploratory phase. This will give us insights for the upcoming part of the code to solve given problem.

Suggestion:

- You may want to plot the sample chosen and get statistics from them, example:

```
import matplotlib.pyplot as plt
import seaborn as sns
samples_for_plot = samples.copy()
samples_for_plot.loc[3] = data.median()
samples_for_plot.loc[4] = data.mean()

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.

Awesome work.

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

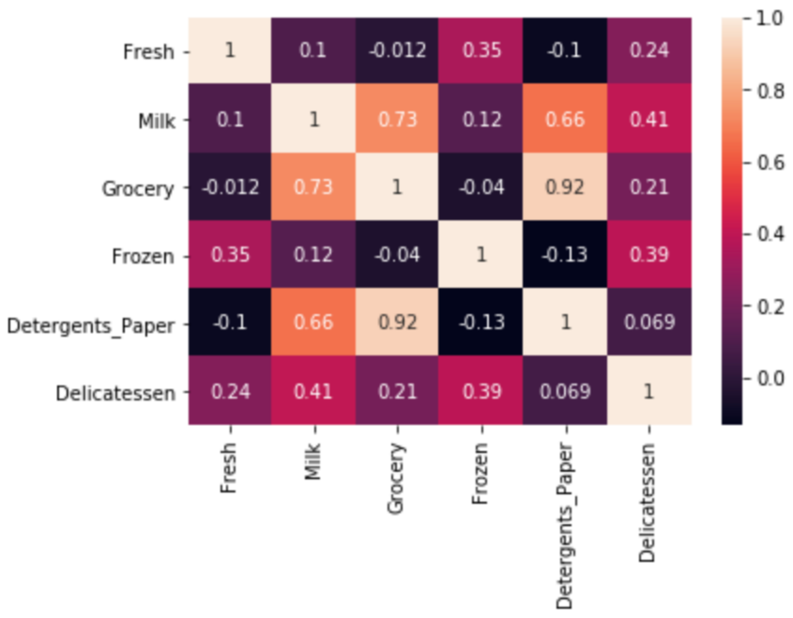
Suggestion:

- You can also plot a heatmap with correlation in numbers to easily identify from the visual, example:

```
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>
```



	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
Fresh	1	0.1	-0.012	0.35	-0.1	0.24
Milk	0.1	1	0.73	0.12	0.66	0.41
Grocery	-0.012	0.73	1	-0.04	0.92	0.21
Frozen	0.35	0.12	-0.04	1	-0.13	0.39
Detergents_Paper	-0.1	0.66	0.92	-0.13	1	0.069
Delicatessen	0.24	0.41	0.21	0.39	0.069	1

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 those outliers and removed them all:

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

Bonus:

- Here is an great article discussing about the strategy of 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.

Very well answered:

> 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.

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

Great job implementing this code:

```
from sklearn.decomposition import PCA
# TODO: Apply PCA by fitting the good data with the same number of dimensi
ons as features
pca = PCA()
pca.fit(good_data)

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

# Generate PCA results plot
pca_results = vs.pca_results(good_data, pca)
```

Bonus:

- Here are great articles about why PCA matters:
  https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32
  https://hackernoon.com/supervised-machine-learning-dimensional-reduction-and-principal-component-analysis-614dec1f6b4c

component-analysis-614dec1f0b4c
https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/

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.

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.

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.

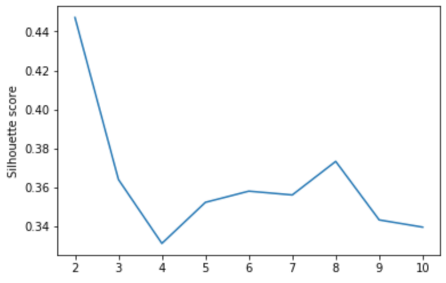
You are right:

> Two clusters will almost always give the best Silhouette score of approximately 0.42 (depending on what outliers were removed).
>
> for n = 2, silhouette score = 0.4467
> for n = 3, silhouette score = 0.3526
> for n = 4, silhouette score = 0.3151
> for n = 5, silhouette score = 0.3135
> for n = 6, silhouette score = 0.3384
> for n = 7, silhouette score = 0.2499
> for n = 8, silhouette score = 0.3199
> for n = 9, silhouette score = 0.3438
> The best silhouette is with n =2

Suggestion:

- You may want to plot the scores and results into a plot for easy visual decision, try it out:

```
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.

Good job.

Conclusion

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

Good job.

Bonus:

- Here I will provide some good articles about A/B testing in AI and how they complement 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 explanation.

> 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.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ