Relational Analysis of Sales Data

Madhavkumar Sheladiya
MAC, University of Windsor
sheladim@uwindsor.ca

Vraj Shah
MAC, University of Windsor
shah3t@uwindsor.ca

Shyamal Thakkar
MAC, University of Windsor
thakka34@uwindsor.ca

***Abstract*— Nowadays, all retail businesses strive to develop the most appealing strategies for persuading people to purchase their goods. As people shop aggressively on days like Black Friday and Cyber Monday, retailers must identify a specific target group and the relationship between them to increase sales. This research will do a relational analysis of the existing Black Friday dataset [1] and offer a pipeline for evaluating any sales data. The clusters groups, as well as the identified co-relation between them, will be included in the results.**

Keywords—retail, customers, analysis, group

I. INTRODUCTION

People nowadays buy aggressively on special days such as Black Friday and Cyber Monday to get great deals. As a result, a variety of retail and other industries are always competing to determine the right target audience for certain products. According to one survey, retail sales increased 14.1 % to $886.7 billion during the holiday season of 2021. Despite the ongoing COVID-19 pandemic, inflationary pressures, and supply chain problems, this was a new record. During the largest five-day holiday shopping period, from Thanksgiving Day to Cyber Monday, about 180 million Americans shopped for various things [2].

It is unquestionably worthwhile to observe and detect the links between client behavior and industry sales. Reliable data sets are required for this study, and even after obtaining accurate sales data, uncovering consumer market patterns requires a certain level of skill. Firstly, the main beneficiaries of such data analysis will be merchants, who will be able to determine the appropriate items to put on sale, and customers, who will be able to purchase the products they desire. Secondly, obtaining accurate estimates can also aid in the development of marketing concepts.

This study proposes the pipeline to assess any sales data and hopes to answer some questions for data analysts to make sense of their data.

II. RELATED WORK

A. Understanding consumer intentions on two major shopping days

In the first paper [3], the authors performed a survey to investigate customer Christmas shopping intentions on the two most popular sales shopping days of the year, Black Friday, and Cyber Monday. To support the theoretical investigation, in this literature [3], the authors came up with some hypotheses based on the factors affecting the selection process by consumers of the channels. Data utilized in the analysis to support the hypotheses is gathered from the business students at a midwestern university by a survey form with gives extra credits to students. As a result of this study [3], Consumers perceived Cyber Monday to be far more practical than Black Friday. Cyber Monday and Black Friday, on the other hand, were relatively similar when employed as a predictor of perceived utility. Both were thought to be useful in holiday gift-giving.

B. Analysis of consumer data on black Friday sales using Apriori algorithm

In the second paper [4], with the help of the Apriori algorithm, the author creates a roadmap for evaluating consumers' online buying behavior. The purpose of this research was to aid further analysis of consumer online shopping behavior, which would assist retailers in developing appropriate marketing strategies for selling their products online, which in turn will further help in developing the economy of the country. The data for this research was collected on Black Friday, November 23, 2018, the day after Thanksgiving. Because the holiday season begins on this day, it is traditionally the busiest shopping day of the year. Since the holiday season accounts for roughly 30% of annual retail sales, it is critical for the economy. In the second paper [4]. The findings of this study provide us with association rules that have the best values for all the above-mentioned attributes, and these rules can be used by business owners to attach those products together in a deal since customers prefer buying them together.

C. Black Friday Sales Prediction and Analysis

In the third paper [5], the main emphasis is on analyzing all consumer data and determining the relationship between independent variables and the target variable, as well as training and testing it to predict expected sales. The goal of this research is to develop a predictor with clear economic interest for store owners, as it will aid them in financial planning, stock management, advertising, and promotions. [5]

The algorithm proposed in the study is the Random Forest regressor, which is a technique that uses multiple decision trees and a technique called Bootstrap Aggregation, also known as bagging, to perform both regression and classification tasks. Instead of relying on individual decision trees, the basic idea is to combine multiple decision trees to determine the final output. The algorithm was performed on the Black Friday dataset [5]. As far as the result of this study is concerned, The Random Forest Algorithm best predicts Black Friday sales with an accuracy of around 81 percent and a root mean squared error score (RMSE) of 2829.09 [5]. Hence, Machine learning techniques, according to the study, produce better prediction models that can be used in stores, and store owners can analyze their customer base to target customers in a better way and increase sales on Black Friday.

III. PROPOSED MODEL

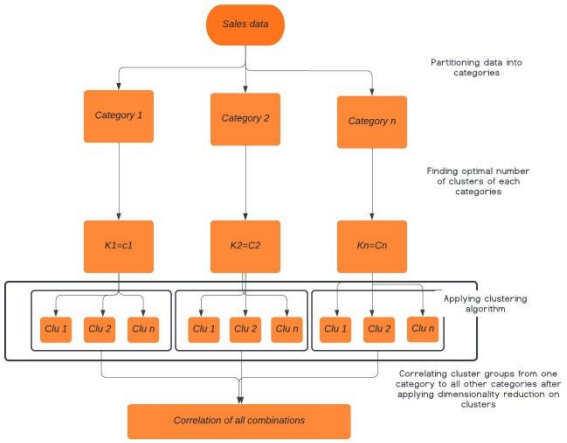
We propose a pipeline for finding relationships between pre-determined categories in any relevant sales dataset or between multiple sales datasets having similar features in between themselves. The datasets must be cleaned and adequately pre-processed into a base form before being applied to the pipeline as shown in fig 1 for example sales data for Black Friday[1].

	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3
0	0	0	10	0	2.0	0	3	9.844506	12.668605
1	0	0	10	0	2.0	0	1	6.000000	14.000000
2	0	0	10	0	2.0	0	12	9.844506	12.668605
3	0	0	10	0	2.0	0	12	14.000000	12.668605
4	1	6	16	2	4.0	0	8	9.844506	12.668605
...	...	...	...	...	...	...	...	...	...
783662	0	2	15	1	4.0	1	8	9.844506	12.668605
783663	0	2	15	1	4.0	1	5	8.000000	12.668605
783664	0	2	15	1	4.0	1	1	5.000000	12.000000
783665	0	4	1	2	4.0	0	10	16.000000	12.668605
783666	0	4	0	1	4.0	1	4	5.000000	12.668605

783667 rows × 10 columns

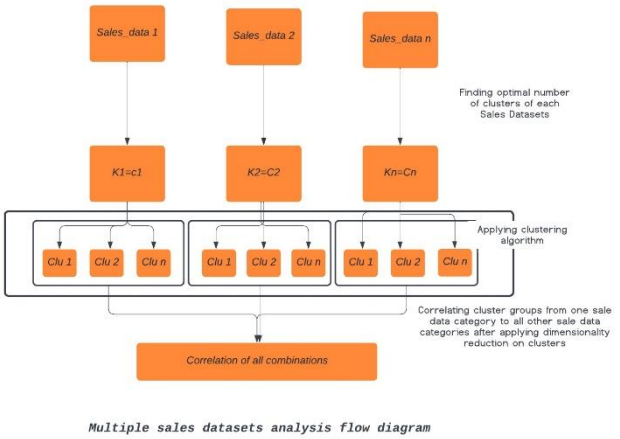
Figure 1: Sample Dataset (cleaned and pre-processed)

The pipeline can be approached in two ways. First, for a single sales data where the dataset can be partitioned into categories strategically as shown in fig 2. And second, for multiple sales datasets where we skip the partitioning and directly apply clustering followed by correlation as shown in fig 3.



Single sales dataset analysis flow diagram

Figure 2: Approach 1 (Pipeline)



Multiple sales datasets analysis flow diagram

Figure 3: Approach 2 (Pipeline)

In the first approach, where there is only a single dataset, we follow the pipeline as seen in fig 2. First, we find the appropriate categories by which the dataset can be partitioned for further analysis. Also, it has to be seen that the dataset is cleaned and pre-processed before to get the best results. Next, we use elbow and silhouette methods to determine the optimal number of clusters for each category. Determining the optimal number of clusters is a crucial step for a y unsupervised algorithm for clustering.

By fitting the model with a range of values for K, the elbow method assists data scientists in determining the ideal number of clusters (K is the number of clusters). If the line chart resembles an arm, the "elbow" (the curve's point of inflection) is a good indicator that the underlying model fits best there. The scoring parameter metric is set to distortion by default, which calculates the sum of squared distances between each point and its allocated center. This is how the score for each value of K is calculated and plotted so that it may be compared to other values to find the best K value.[6] When the ground truth about the dataset is unknown, the Silhouette Coefficient is utilized to compute the density of clusters computed by the model. The silhouette coefficient for each sample is calculated by averaging the difference between the average intra-cluster distance and the mean nearest-cluster distance for each sample, normalized by the maximum value. This generates a score ranging from 1 to -1, with 1 indicating highly dense clusters and -1 indicating completely inaccurate grouping.[7] We use elbow and silhouette methods as instead of using a single algorithm to determine an optimal number of clusters use of two algorithms that complement each other's result gives out a better result and can be considered safer.

Now, as the optimal number of clusters is determined the next step would be to form cluster groups within each category. Here we use the k-means algorithm to determine cluster groups for each category. The k-means algorithm although being primitive is a very robust algorithm and is one of the most widely used algorithms for clustering problems. Either Lloyd's or Elkan's algorithms are used to tackle the k-means problem. O (k n T) is the average complexity, where n is the number of samples and T is the number of iterations. With n = number of samples and p = number of features, the worst-case complexity is O(n(k+2/p)) as calculated by D. Arthur and S. Vassilvitskii in their research on "How slow is the k-means method?" SoCG2006. The k-means technique is

highly quick in reality (one of the fastest clustering algorithms known), but it tends to fall into local minima. That's why restarting it numerous times can be beneficial. [8]

Next, as we have the cluster groups for each category, we need to apply correlation on top of those clusters. To achieve this the clusters, need to be reduced in dimensions and fit into a matrix of each category where the rows define the cluster vector of the category, and the columns define the features for the category. For dimensionality reduction on clusters, we use a truncated SVD (singular value decomposition) algorithm. To achieve dimensionality reduction using a truncated SVD algorithm the transformer uses truncated singular value decomposition (SVD) to reduce linear dimensionality. This estimator, unlike PCA, does not center the data prior to generating the singular value decomposition. This means it can efficiently operate with sparse matrices. It offers two algorithms: a fast randomized SVD solver and a "naive" method that employs ARPACK as an eigensolver on X * X.T or X.T * X. [9] For our calculations we have made use of the faster randomized SVD solver.

Now that we have the matrix for each category representing the cluster groups, we apply correlation on those matrices to determine relational similarities between the cluster vectors between different categories within the sales data. We have used Pearson correlation where we calculate the Pearson correlation coefficient and the p-value for testing non-correlation. The linear link between two datasets is measured by the Pearson correlation coefficient. For Pearson's correlation to work, each dataset must be normally distributed. This correlation coefficient, like others, ranges from -1 to +1, with 0 denoting no correlation. A linear relationship is implied by a correlation of -1 or +1. Positive correlations imply that as x rises, y will rise as well. Negative correlations indicate that as x rises, y falls. The p-value roughly represents the likelihood that an uncorrelated system can produce datasets with a Pearson correlation at least as extreme as the one calculated from these datasets. The p-values aren't completely dependable, but they're probably fine for datasets of 500 or more. [10]

At last, the results from the correlation can be charted in a heatmap to analyze and make sense of the results. If the dataset has a possibility to be categorized with the use of different parameters, then it is highly recommended to try the pipeline proposed in multiple settings to get the best read from the results on that dataset.

In the second approach, where there are multiple sales datasets, we follow the pipeline as seen in fig 3. Here, as we already have multiple sales datasets and we want the correlation between those datasets, we can skip the categorizing part of the first approach and directly jump to determining the optimal number of clusters for each sales data. So, for the second approach we replace the categories with multiple sales datasets and from there follow the same process as done for the first problem.

IV. Results

To demonstrate our pipeline, we have made a sample run following approach 1 shown in fig 2 on the Black Friday sales dataset from kaggle [1]. This dataset consists of very similar readings uniformly distributed between the parameters and the categories within the datasets. So, having a dataset with such properties will be very useful when analysing the results.

For our dataset, we first categorize it on the basis of cities and apply elbow and silhouette methods to find the optimal number of cluster groups (k) to form for each category. As we can see in fig 4, 5, and 6, for city A and city B we can form 3 cluster groups, and for city C we can form 4 cluster groups to better define their respective categories. From fig 4, 5, and 6, we consider both the elbow from the elbow method and the spike in the graph plus the uniformity of the cluster distributions and how close they are to their silhouette score from the silhouette method to determine the optimal number of clusters (k).

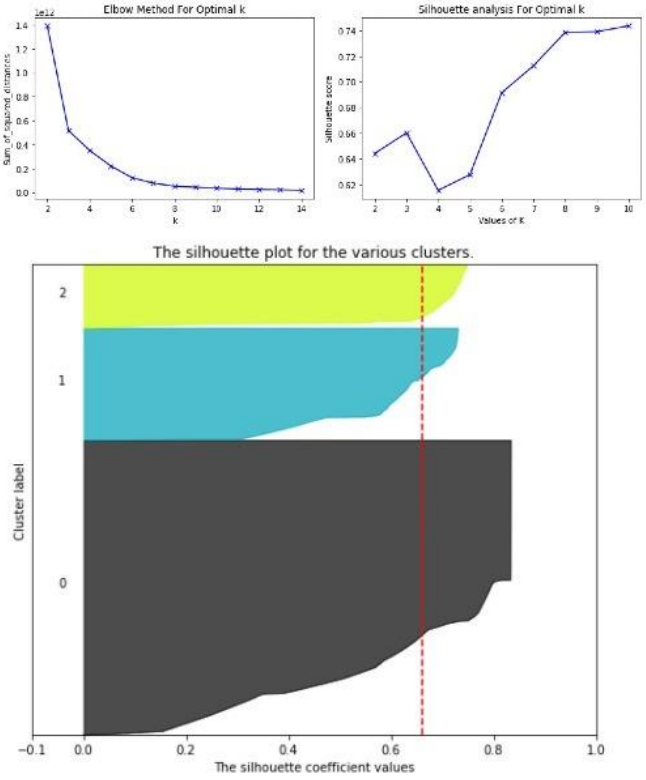


Figure 4: Determining k for category A

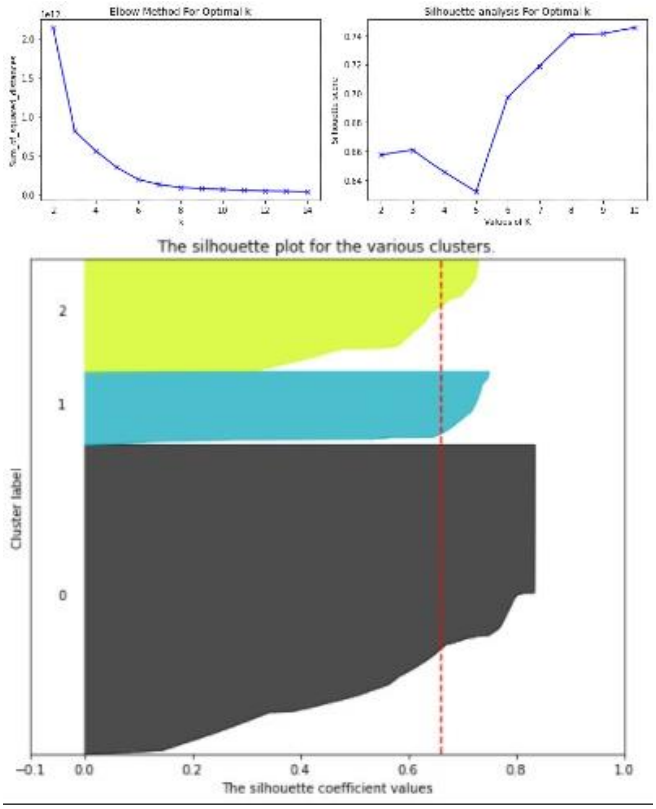


Figure 5: Determining k for category B

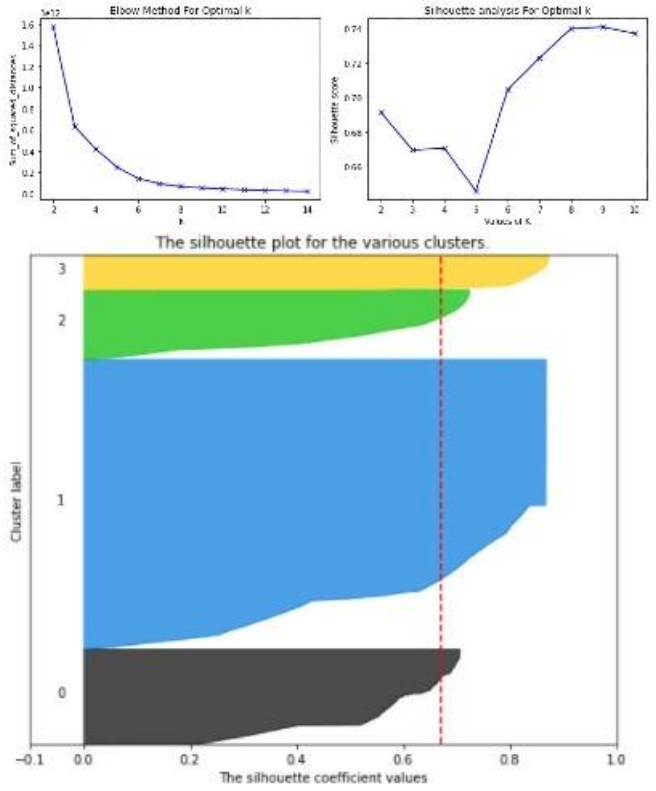


Figure 6: Determining k for category C

Next, we apply the k-means algorithm to each category to obtain cluster groups for each category. Now that we have the clusters, we for a matrix of clusters for each category by reducing the dimensions of cluster matrices using truncated SVD. We can see how the information is stored in the matrix from fig 7 for categories A, B, and C respectively.

A

```
[array([[4.55988409e-05, 1.30288802e-04, 4.59678553e-04, 1.03441887e-04,
        2.08532657e-05, 2.25778773e-04, 5.03935744e-04, 7.26344523e-04,
        9.99999463e-01]]),
 array([[1.56309159e-04, 4.73285467e-04, 1.62605353e-03, 3.80195488e-04,
        8.18141327e-05, 1.37265995e-03, 2.23868989e-03, 2.65227229e-03,
        9.99991513e-01]]),
 array([[8.08258116e-05, 2.45554091e-04, 8.34841528e-04, 1.93313431e-04,
        4.12549037e-05, 5.59047318e-04, 1.05172879e-03, 1.35594450e-03,
        9.99997970e-01]])]
```

B

```
[array([[8.02177808e-05, 2.65837214e-04, 8.57732567e-04, 2.00508128e-04,
        4.36394470e-05, 5.46181279e-04, 1.04316037e-03, 1.35341957e-03,
        9.99997963e-01]]),
 array([[1.50682824e-04, 5.02372632e-04, 1.61581559e-03, 3.86154898e-04,
        8.41868075e-05, 1.37607691e-03, 2.22667603e-03, 2.64289563e-03,
        9.99991561e-01]]),
 array([[4.64170090e-05, 1.44294775e-04, 4.76916411e-04, 1.09986375e-04,
        2.37331808e-05, 2.10377918e-04, 5.01324353e-04, 7.30743826e-04,
        9.99999454e-01]])]
```

C

```
[array([[8.17915844e-05, 3.03001080e-04, 9.24835851e-04, 2.07544862e-04,
        4.84555040e-05, 5.74570853e-04, 1.09775158e-03, 1.39390912e-03,
        9.99997761e-01]]),
 array([[4.07146111e-05, 1.35478401e-04, 4.31392124e-04, 9.46624031e-05,
        2.17370022e-05, 2.01440222e-04, 4.51815197e-04, 6.50413168e-04,
        9.99999558e-01]]),
 array([[1.53564866e-04, 5.59432702e-04, 1.73062608e-03, 3.88546286e-04,
        8.80329485e-05, 1.41753458e-03, 2.25787743e-03, 2.65329195e-03,
        9.99991181e-01]]),
 array([[5.39823624e-05, 1.80672856e-04, 5.84616878e-04, 1.27704240e-04,
        2.93077020e-05, 2.08490728e-04, 5.53688067e-04, 8.48622922e-04,
        9.99999268e-01]])]
```

Figure 7: Matrix for all categories

After forming the matrices for all the categories, we correlate each cluster vector from one category with all cluster vectors from other categories covering all combinations. To correlate those vectors, we use Pearson correlation and plot the values in the form of a heatmap to better visualize and analyse the results. We can see the output from the Pearson correlation in fig 8 for correlations between AB, AC, and BC categories respectively.

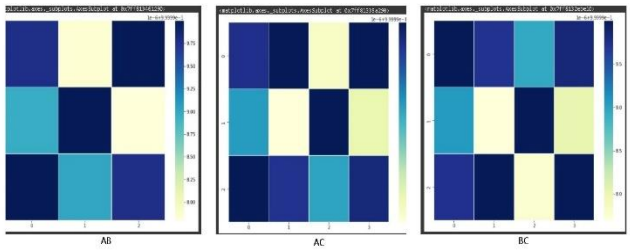


Figure 8: Correlation results

From the heatmap, we can observe the similarities between the cluster vectors from different categories. In the heatmap, the darker blocks show high correlation, and the lighter blocks no to very less correlation between cluster vectors. As we knew that our sample dataset had uniform distribution generally, we get results with a high amount of correlation between cluster vectors and hence between categories. This kind of output can be very useful to analyze categories. For an example case, if the city B is a small city and has a very low-risk factor, knowing that city A and C have high correlations, any kind of new strategies or experiments can be done on city B and the results from those can be expected to go similar in city A and C. This can also be assumed true for the single cluster group from city B to correlating cluster groups from city A and C.

V. Limitations

Firstly, if the pipeline is being applied on single sales dataset than the dataset should be categorizable. Secondly, if multiple sales datasets are being used on the pipeline the features or the parameters should be similar between those datasets for the results to make sense. Lastly, although we have shown an example using primitive algorithms to explain our pipeline using more refined algorithms could generate better results

VI. Conclusion and Future work

There always is a need to find meaningful information out of data to profit from it. Also, data analysts are always in search of new and proven ways to extract this meaningful information out of data. Having a new way to extract information means they are ahead of the competition. The proposed pipeline will help data analysts to extract information out of their sales dataset and visualizing the result will help in deducing better conclusions. The pipeline will also act as a template for data analysts to follow so that they can have a path to follow to relationally analyse the sales data. Hence, turn their sales data into information that can help them to come up with new ideas to make a profit. This will help companies profit from new ideas with a lower risk factor.

Although the results seem conclusive the pipeline in its core works with primitive algorithms. The use of more advanced algorithms in the pipeline could benefit the pipeline in generating even more clear results. The pipeline can also be branched out to follow different ideas instead of clustering or correlating to a new path which is not researched in this paper. Also, combining this pipeline with some other existing pipelines to see if that gives better results is an interesting way to look forward. There are many such pathways in which the proposed pipeline can be refined in the future.

References

[1] StefanDolezel. Black Friday. Kaggle.com. Published 2018. Accessed April 5, 2022. https://www.kaggle.com/datasets/sdolezel/black-friday

[2] Spendmenot.com. Published 2022. Accessed April 5, 2022. https://spendmenot.com/blog/black-friday-sales-statistics/

[3] Swilley, E., & Goldsmith, R. E. (2013). Black Friday and Cyber Monday: Understanding consumer intentions on two major shopping days. Journal of retailing and consumer services, 20(1), 43-50.

[4] Maharjan, M. (2019). Analysis of consumer data on black friday sales using Apriori algorithm. SCITECH Nepal, 14(1), 17-21.

[5] Reddy, P. R., & Sravani, K. (2020). Black Friday Sales Prediction and Analysis. Think India Journal, 22(41), 59-64.

[6] "Elbow Method — Yellowbrick v1.3.post1 documentation," Scikit-yb.org, 2021. https://www.scikit-yb.org/en/latest/api/cluster/elbow.html (accessed Apr. 05, 2022).

[7] "Silhouette Visualizer — Yellowbrick v1.3.post1 documentation," Scikit-yb.org, 2021. https://www.scikit-yb.org/en/latest/api/cluster/silhouette.html (accessed Apr. 05, 2022).

[8] "sklearn.cluster.KMeans," scikit-learn, 2022. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (accessed Apr. 05, 2022).

[9] "sklearn.decomposition.TruncatedSVD," scikit-learn, 2022. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html (accessed Apr. 05, 2022).

[10] "scipy.stats.pearsonr — SciPy v0.14.0 Reference Guide," Scipy.org, 2014. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html (accessed Apr. 05, 2022).