**COVID-19 Open Research Dataset Challenge (CORD-19):**

**Using Machine Learning to Identify Potential COVID-19 Risk Factors Using Research Studies**

Joshua Harkness & Justin Kennedy

[**https://github.com/darkhark/COVID**](https://github.com/darkhark/COVID)

***Abstract* - The purpose of this research was to apply modern machine learning and data mining techniques to categorizing scientific literature in a way that allowed potential risk factors to the transmission and infection of COVID-19 to be identified.  Currently there is a wealth of literature (and more every day) regarding this topic, but this rush of new information has made it difficult for scientists to keep up with the research. Our research included data preparation and cleansing, data clustering and visualization using multiple techniques, including k-means, k-folds and t-distributed Stochastic Neighbor Embedding (TSNE), followed by the interpretation of the results.  The ultimate goal of identifying new or unique risk factors for COVID-19 by analyzing clustered scholarly articles was not able to be fully accomplished, however this investigation did yield a promising foundation by successfully clustering the literature and identifying which clusters deserve future exploration.**

1. Introduction

When a new virus presents itself as a global threat, knowledge is the most important weapon in fighting it’s spread.  Our global connectivity presents both a boon and a challenge to efficient knowledge accumulation due to the sheer volume of data available.  Luckily, machine learning techniques exist to help comb through that data and separate the signal from the noise. However, in machine learning is no “one-size-fits-all” solution.  This paper explores different machine learning and data mining techniques to identify and categorize potential risk factors related to COVID-19, using currently available research and epidemiological studies. Previous efforts on categorizing or classifying text within the Natural Language Processing (NLP) field have established certain methodologies as the standard for any project.  These include clustering methods such as k-means and Latent Dirichlet Allocation (LDA), and data dimensionality reduction methods such as Latent Semantic Analysis (LSA) and TSNE. Unfortunately, each data set presents its own unique NLP challenges and requires a bespoke solution to be built. The COVID-19 outbreak resulted in numerous research articles that may contain insights to potential risk factors associated with COVID-19 but are impossible to find using manual methods.  Developing a custom literature clustering solution that identifies these risk factors, along with identifying potentially novel risk factors as they emerge, would allow doctors and medical professionals to most efficiently treat and prevent transmission of COVID-19.

1. Materials and Methods
2. *Computer Language and Hardware/Software Systems*

The computing environments used for data preparation, exploration and statistical learning experimentation were personal laptop computers with hardware and software configurations as detailed in APPENDIX, Table 4. The Python© programming language, version 3.7.6, in conjunction with the Spyder©, Anaconda© and JetBrains© PyCharm© Integrated Development Environments (IDE) was used for data cleansing, preparation and analysis. Results were visualized using Spyder© and PyCharm©.

1. *Dataset*

The kaggle.com COVID-19 Open Research Dataset (CORD-19) was used to evaluate if a classification and clustering model could be developed to help identify potential risk factors listed in the scholarly articles contained therein.

The full dataset today contains over 51,000 articles, including over 40,000 with full text, concerning COVID-19, SARS-CoV-2, and other related coronaviruses. The dataset was periodically updated; however the dataset was utilized for this investigation was frozen on the update that occurred on 3 April, 2020. This dataset contained approximately 47,000 articles and approximately 36,000 articles that contained full text. The dataset was originally downloaded from kaggle.com and contained several types of files. The types of files and their descriptions can be seen in Table 1:

**Table 1 - File Names and Descriptions**

|  |
| --- |
| **File/Folder Name Description** |
| json\_schema.txt JSON schema of full text documents  metadata.csv Metadata for 47,000 articles  metadata.readme Readme file for metadata.csv  biorxiv\_medrxiv Folder containing 1,342 research article JSON objects  comm\_use\_subset Folder containing 9,365 common-use research article JSON objects custom\_license Folder containing 23,152 custom-license research article JSON objects  noncomm\_use\_subset Folder containing 2,377 non-common-use research article JSON objects |

The finalized dataset that was used for analysis consisted of a PANDAS dataframe object, meta\_df, which contained all of the metadata from metadata.csv, and a list, all\_json, which contained all of the JSON files present in the original dataset. The attribute *paperID*, which was common between the original JSON object containing the article and each entry in metadata.csv, was used as the linking identity between the two. Each row in metadata.csv contained high level information about each original JSON, such as author, title, publisher and abstract, and was used to determine if the individual JSON was loaded into the all\_json list. The final processed list, all\_json, had a length of 52,097 entries, which was cut down into a resultant dataframe, post loading process, of 32,417 entries. Each contained the minimum elements *paperID, abstract, body\_text, authors, title,* and *abstract\_summary.*

1. *Data Cleansing*

The obvious issue of having a large number of entries, combined with a naked eye review of the dataset, revealed that an extensive amount of data preparation and cleansing would be required to ensure the follow-on model testing and analysis was as efficient and accurate as possible. This common NLP problem required the input data to be formatted and standardized to an acceptable level. Fields that lacked data required additional attention. Ultimately, the entries that were missing an abstract or abstract summary were retained and replaced with the placeholder text *“No abstract provided”*.

While having an abstract was desirable, the lack of abstract content was not considered to be an indicator of a potentially low-quality entry. A different approach for the authors and content within the field *body\_text* was taken. During the initial loading process, if it was detected that the entry was missing either *author­* or *body\_text*, that entry was determined to lack credibility and usefulness and was not appended to the dataframe. Following the initial loading of the data, all non-English entries were removed. Attempting to include non-English language articles could have posed a problem for later clustering efforts due to software-based translations not properly carrying over the proper semantic context, which is especially important when analyzing medical journals. Additionally, the number of non-English articles were minimal, so the easiest method of handling them was removal. This reduced the dataframe to 31,515 entries.

Next, the text of these entries was standardized for NLP processing. This involved detecting and removing any potential duplicate articles as well as cleansing the remaining text entries by removing stopping words, punctuation and converting the entire dataframe to lower case. The step-by-step approach that was taken to data cleansing is listed below in Figure 1:

**Figure 1 - Data Cleansing Process**

1. *Data Preparation*

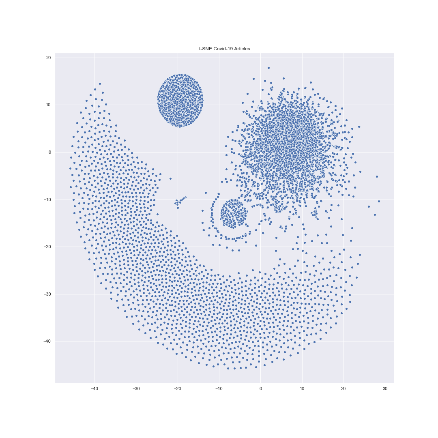
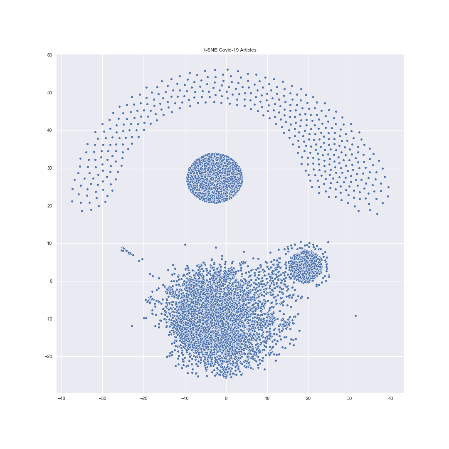
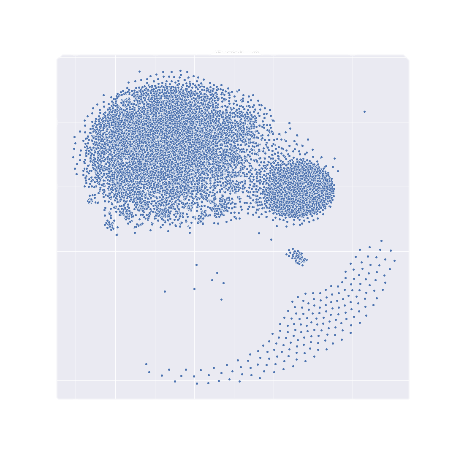
Once the dataframe was fully populated with the cleansed data, the next step was to prepare the data for clustering. The first action taken was to take the abstract and body text entries, which were stored as one string for the entire entry and convert them into something that could be used by future algorithms. This was accomplished by using Term Frequency-Inverse Document Frequency (TF-IDF) on the individual words, which were created by splitting the strings on spaces. TF-IDF assigned a score to each word within a single document based on its frequency within that document. This was then measured against the frequency of that word in all other documents in the dataset and the score was reduced for how frequently that word repeated itself in the entire corpus of text analyzed. More simply put, it discovered which words were important to a single document based on its frequency within that document and how rare the word was overall. There are several other methods other than TD-IDF, however, according to a study published in 2016 [1], TF-IDF was the most popular method and it was decided to select that method.

After selecting TF-IDF as the method of vectorization for the data’s abstract and body text, TF-IDF was separately applied to both the body and abstract vectors. Since the abstract contained a summary of the paper, the anticipation was that there would be a concentration of potential risk factors within the abstract text. As the abstract texts contained an average of approximately 300 words, and the body texts contained an average of approximately 4,500 words, the hope was that by analyzing the abstract instead of the body valuable risk information could be identified by utilizing a smaller, more efficient set of data and reduce computation times.

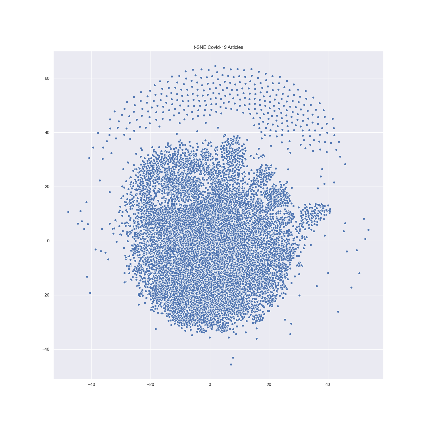
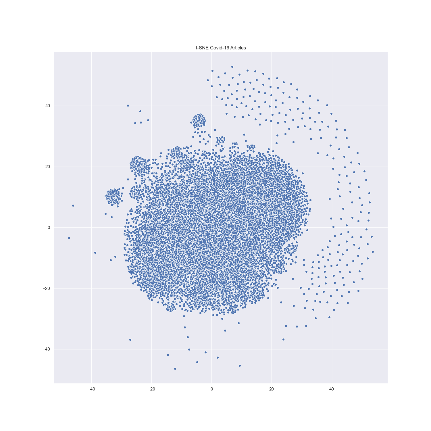
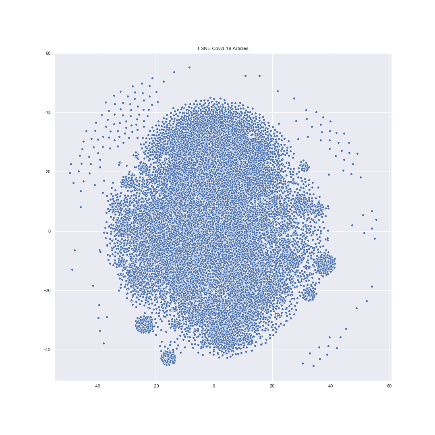
1. *N-Grams*

The first attempt at identifying clusters within the literature focused on utilizing n-grams to capture potential bigrams or trigrams that, when clustered and analyzed, could signify potential articles related to risk factors associated with COVID-19 susceptibility or mortality rates. After passing both the body and abstract through the TD-IDF process, experiments were ran on separating each corpus into bigrams or trigrams. Finally, t-Distributed Stochastic Neighbor Embedding (TSNE) was utilized for visualizing the end results, where a *“k”* argument was passed to the TSNE argument for it to use as its k-nearest-neighbor (KNN) value. Many combinations of bigrams and trigrams detections on multiple *k* values for the KNN algorithm were run. The end results of those runs are shown below in Figures 2 and 3:

**Figure 2 - Abstract TSNE w/ 25, 15, and 5 Neighbors**



**Figure 3 - Body TSNE w/ 25, 15, and 5 Neighbors**



As is evidenced by the images, the clustering efforts were partly successful on both the body and the abstract, but the clusters were not distinguishable. In conclusion, it was determined that the n-gram separation combined with KNN clustering in TSNE was not able to reliably delineate clusters based on common topics. Additionally, it was concluded that because the abstract data resulted in distinct, but few, clusters that it would not be a sufficiently diverse enough data set to run an analysis on and was subsequently dropped from future clustering analysis. Upon further investigation, it was determined that the small and distinct clustering was caused by a sizeable minority of abstract entries initially containing zero text and subsequently being filled with identical placeholder text.

1. *PCA/K-Means Clustering*

After realizing that the n-grams approach was insufficient, Principal Component Analysis (PCA) combined with K-Means clustering was utilized. PCA is a dimension reduction algorithm and K-Means is a method that attempts to partition the observations into *k* clusters by minimizing within-cluster variances. The intent behind utilizing both was that PCA would reduce the number of dimensions so that the follow-on K-Means effort was more efficient. After establishing that K-Means was the next clustering method, the optimal *k* needed to be found for the algorithm. To accomplish this, the Elbow Method was initially utilized, and when that proved insufficient by itself, the Silhouette Method was added for verification (see below in Figure 4).

**Figure 4- Elbow and Silhouette Results for Determining Optimal K**

A picture containing man, water, table, people

Description automatically generated A picture containing text

Description automatically generated

The Elbow Method is probably the most well-known method for determining the optimal number of clusters [2]. However, for data that is not clearly clustered, such as the dataset used, the method may not result in an elbow that is easy to delineate. Based on the Elbow Method, the optimal *k* value could be anywhere between approximately 12 and 19. When the Elbow Method proves insufficient or provides ambiguous results (like the result presented here), the Silhouette Method is often used to augment or validate suppositions made by using the Elbow Method. The main focus of the Silhouette Method is to provide a measure of how similar a point is to its own cluster compared to other clusters [2]. The range of the Silhouette value is between -1 and 1, with a value close to 1 being desired. As seen in Figure 4, the *k* values that possessed the highest Silhouette value were less than 10. However, it did appear a local optima *k* value around *k* = 16 existed. This corresponded well with our Elbow Method supposition of *k* being between 12 and 19. Ultimately, a value of *k* = 16 was utilized for K-Means clustering.

Once the optimal *k* value was identified, the analysis proceeded by providing the vectored data through a PCA algorithm followed by the K-Means algorithm. Once the K-Means algorithm identified the desired 16 clusters, TSNE was then used to produce a visualized output of the results by reducing the high dimensional data down to two-dimensional x and y coordinates. As seen in Figure 5 (below), multiple distinct clusters were identified within the literature that could be used as a basis to begin an analysis of topic and key word importance.

**Figure 5- K-Means Clustering Results**

A close up of a logo

Description automatically generated

As a final validation of the clustering, a Stochastic Gradient Descent (SGD) algorithm was utilized to build a model and test for accuracy. The matrix from the TF-IDF vectorization was ingested, with the cluster number applied by the K-Means used as the dependent variable. After the model was built, it was tested against the data the SGD algorithm was given to train to ensure that the model was not overfit. Once the model was validated as not overfitting the training data, the model applied to the test data to validate accuracy. The results of the SGD test can be seen in Table 2 (below).

**Table 2 - SGD Testing Results**

|  |  |  |
| --- | --- | --- |
|  | Training Dataset | Testing Dataset |
| Size | 25,311 | 6,238 |
| Accuracy | 90.69% | 87.34% |
| Precision | 92.02% | 89.91% |
| Recall | 92.02% | 88.51% |
| F1 | 92.02% | 89.09% |

Having validated the K-Means clustering effort as accurate, efforts began on analyzing the resultant clusters for topic and key-word importance.

1. *Cluster Analysis for Topic and Key-Word Importance*

With the clustering being completed and validated, a disease lexicon was built using the CDC.gov website [3] and a custom-built data cleansing method that was derived from earlier efforts in cleaning the original dataset. This resulted in a list object containing 990 distinct disease names and 956 unique words within those diseases.

With the disease list built, the most impactful keywords in each cluster were identified using Latent Dirichlet Allocation (LDA). LDA reduced the dimensionality of each cluster down to the specified number. In this study, the desired number of top words per cluster was 20. LDA created fuzzy, or probabilistic, clusters based on the words in the bodies that were most probable to occur between the articles. Some clusters did not possess as many high probable, or impactful words and did not reach the desired 20 words.

The top-20 most impactful words from each cluster were put into its own list and a comparative algorithm was applied to find common occurrences between the disease name list and the list containing the top-20 most impactful keywords in each cluster. The results of this comparison can be found in Table 3 (below).

**Table 3- Disease Key-Word Overlap Among Top-20 Most Frequent Words, by Cluster**

|  |  |  |
| --- | --- | --- |
| **Cluster** | **Matched Key Words** | **Number of Overlaps** |
| 1 | ‘viral', 'coronavirus', 'dengue', 'detection', 'viruses', 'virus', 'data', 'bacterial' | 8 |
| 2 | ‘animals', 'network', 'ace', 'pigs', 'water', 'viruses', 'diseases', 'disease', 'virus', 'group', 'blood', 'data', 'use', 'epidemic', 'influenza', 'infection' | 16 |
| 3 | ‘lung', 'cancer', 'treatment', 'asthma', 'hiv', 'chest', 'resistance', 'sars', 'adenovirus', 'diarrhea', 'group', 'malaria', 'pneumonia', 'data', 'surveillance', 'infections', 'infection' | 17 |
| 4 | ‘animals', 'treatment', 'pigs', 'coli', 'bovine', 'disease', 'virus', 'antimicrobial', 'respiratory', 'diarrhea', 'infection' | 11 |
| 5 | ‘risk', 'international', 'vaccines', 'emergency', 'diseases', 'human', 'response', 'virus', 'testing', 'global', 'vaccine', 'data', 'surveillance', 'influenza' | 14 |
| 6 | ‘animals', 'viral', 'ace', 'detection', 'viruses', 'human', 'bovine', 'disease', 'immune', 'virus', 'brain', 'data', 'cell', 'infection' | 14 |
| 7 | ‘dogs', 'ace', 'winter', 'data', 'influenza' | 5 |
| 8 | ‘orf', 'immune', 'group', 'vaccine', 'cell', 'infection' | 6 |
| 9 | ‘lung', 'viral', 'vaccines', 'treatment', 'human', 'disease', 'virus', 'liver', 'skin', 'group', 'vaccine', 'mouse', 'leptospirosis', 'infection' | 14 |
| 10 | ‘viral', 'ace', 'detection', 'yeast', 'hiv', 'virus', 'sars', 'data', 'influenza' | 9 |
| 11 | ‘viral', 'treatment', 'dogs', 'common', 'pandemic', 'diet', 'group', 'blood', 'data', 'cell', 'influenza' | 11 |
| 12 | ‘viral', 'pigs', 'group', 'farm', 'infection' | 5 |
| 13 | ‘lung', 'viral', 'treatment', 'stress', 'human', 'disease', 'hiv', 'immune', 'virus', 'bacterial', 'mouse', 'infection' | 12 |
| 14 | ‘lung', 'viral', 'orf', 'human', 'data' | 5 |
| 15 | ‘children', 'hpiv', 'evd', 'detection', 'asthma', 'birth', 'disease', 'rsv', 'immune', 'pneumonia', 'vaccine', 'bronchiolitis', 'data', 'bacterial', 'influenza' | 15 |
| 16 | ‘animals', 'bird', 'cats', 'treatment', 'dogs', 'horses', 'avian', 'rabies', 'human', 'intestinal', 'virus', 'liver', 'diarrhea', 'guinea', 'blood', 'cell' | 16 |

A manual review of this indexing shows that clusters 3, 5, 6, 13,15 and 16 show promise for future investigation and extrapolation in order to identify research articles that contain potential risk factors for COVID-19.

1. Discussion

This investigation presented several challenges during each phase. Broadly, NLP problems are difficult due to the inherent difficulties in attempting to get computer algorithms to understand the semantic intent behind the written word. Piled on top of this was the decision to participate in an investigation that involved a developing dataset that examined a new viral outbreak. A significant amount of time was spent during the data preparation and data cleansing phase. This phase continued throughout the course of the investigation as new outliers were discovered and were required to be handled. As an example, the removal of non-English language texts did not occur until after the several rounds of n-grams and KNN clustering.

The sheer size of the data was also a challenge that required multiple iterations to overcome. During the initial phases of the investigation the original data was cleansed and prepared, and the resultant dataframes constructed, each time a new experiment was run. Eventually a solution was implemented in which the data was cleansed and prepared and the output was written to a .csv file for faster future loading. This was not the only time-intensive aspect of the investigation that needed to be overcome. The clustering efforts beginning with n-grams and KNN required extensive testing as results were recorded and optimizations were made. After moving on from n-grams and KNN clustering to PCA and K-Means, the investigation finally reached a point where the clustering had been validated as accurate. It was finally at this point that the models and vectorized matrix could be saved for future utilization, thereby saving considerable amounts of time through Python’s pickles. The size of the dataset also presented a problem not only from the standpoint of the time it took to execute, but from the standpoint of pure feasibility. For example, during the SGD testing and validation, the last step was to obtain a KFolds mean average score from the entire dataset all at once, instead of simply comparing the data in the split. This proved to be too computationally intensive for the personal laptops used for this investigation and subsequently ran out of memory and failed.

After running the cross-indexing of diseases with most frequent words within clusters, there were several that showed promise for future exploitation. Unfortunately, time remained the ultimate constraint and prevented this exploration to be accomplished.

1. Future Work

The outcome of this investigation yielded several critical components for future work to continue. These include the ability to ingest and clean various forms of scientific literature into a standardized dataframe, the ability to retain and export the components required for clustering in order to speed up future iterations and a method for determining the accuracy of the clustering efforts to ensure that follow on analysis would be accurate. This does leave one significant component untouched by this investigation: the ability to identify research articles that contain information related to potential risk factors associated with COVID-19. This is the most significant component of this problem and will require an extensive effort to optimize. This investigation was only able to reach the point of utilizing a lexicon of diseases and index that against the occurrence of those words among the top 25 most impactful words within each cluster. This was a step in the right direction but not enough to begin addressing the true aim of this investigation, which was identifying potential risk factors associated with COVID-19.

After seeing the initial results of the key-word overlap, it was decided that a future investigation would benefit from applying the same concept to the filtering of the initial text size, meaning applying the concept of taking a lexicon and finding a threshold of desired overlap on the analyzed texts and removing all other extraneous words. Another aspect that could be improved would be accessing cloud-based computing resources to help augment an investigation’s ability to run the more intensive tasks, such as the SGD model and verification, along with running any future improvements applied to the existing techniques of clustering.

1. Conclusion

COVID-19 has swept the globe and become a pandemic of potentially historical proportions. As with any emergent disease, information about risk factors, treatments and cures are questionable during the initial phases. This is partially caused by the sheer volume of research and medical literature that is produced during these initial phases. If there was a way to more accurately and quickly organize this literature, it would save medical professionals time, and ultimately, lives. The work provided during this investigation demonstrates that it is possible to apply modern machine learning techniques to accurately group research articles into common clusters. What remains to be validated is if these clustered articles could have additional NLP methods applied to them to tease out valuable information regarding risk factors, treatments or other palliative measures. A model that can accomplish what has been described would be truly valuable on a global scale.

1. References

[1] *"Research-paper recommender systems: a literature survey"*, 26 July 2016.  Breitinger, Corinna; Gipp, Bela; Langer, Stefan. [Online]. Available: <https://link.springer.com/article/10.1007/s00799-015-0156-0> [Accessed: 24-Apr-2020].

[2] “*How to determine Optimal K for K Means?”*, 17 June 2019. Mahendru, Khyati. [Online]. Available: <https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb> [Accessed: 24-Apr-2020].

[3] “*Disease & Conditions A-Z Index”,* [Online]. Available: <https://www.cdc.gov/diseasesconditions/az/a.html> [Accessed: 24-Apr-2020].

[4] Lettier, “Your Guide to Latent Dirichlet Allocation,” Medium, 31-May-2019. [Online]. Available: https://medium.com/@lettier/how-does-lda-work-ill-explain-using-emoji-108abf40fa7d. [Accessed: 24-Apr-2020].

VII. Appendix

**Table 4- System Hardware and Software Information for Model Exploration and Selection**

|  |  |  |
| --- | --- | --- |
|  | **System 1** | **System 2** |
| OS Name | Microsoft Windows 10 Pro | Microsoft Windows 10 Pro |
| Version | 10.0.18362 Build 18362 | 10.0.18363 Build 18363 |
| OS Manufacturer | Microsoft Corporation | Microsoft Corporation |
| System Manufacturer | Microsoft Corporation | Dell Inc. |
| System Model | Surface Pro 4 | Precision 5530 |
| System Type | X64-based PC | x-64 based PC |
| Processor  BIOS Version/Date | Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz, 2496 Mhz, 2 Core(s), 4 Logical Processor(s) | Intel® Core™ i5-8400H CPU @ 2.50 GHz, 2496 Mhz. 4 Cores, 8 Logical processors |
| SMBIOS Version | Microsoft Corporation 108.2706.768, 4/18/2019 | Dell Inc. 1.8.1, 2/1/2019 |
| Embedded Controller Version | 255.255 | 255.255 |
| BIOS Mode | UEFI | UEFI |
| BaseBoard Manufacturer | Microsoft Corporation | Dell Inc. |
| Platform Role | Slate | Mobile |
| Locale | United States | United States |
| Hardware Abstraction Layer | 10.0.18362.752 | 10.0.18362.752 |
| Installed Physical Memory (RAM) | 8.00 GB | 8.00 GB |
| Total Physical Memory | 7.93 GB | 7.74 GB |
| Available Physical Memory | 2.00 GB | 1.27 GB |
| Total Virtual Memory | 21.4 GB | 22.9 GB |
| Available Virtual Memory | 9.96 GB | 5.41 GB |
| Page File Space | 13.5 GB | 15.2 GB |
| Integrated Development Environment/Tools | PyCharm® Professional 2020.1 | Spyder® 4.1.2 |
| Language | Python® 3.7.6 | Python® 3.7 |