A dataset with 201 columns and 200,000 rows is provided with unknown target. The primary objective of this project is to identify the columns that are sufficient to capture the "essence" of the entire dataset. Therefore, the provided dataset is scrutinized with various statistical tests in order to isolate out the redundant columns. Successful reduction of the dataset to lower dimensionality can reduce the training time of machine learning models, reduce memory and disk space requirements, and when reduced to 3 or less dimensions, enable visualization of the dataset.

Methodology:

Exploratory Data Analysis:

The first step towards finding patterns in the data is to try to understand the raw data. The large size of the data makes it difficult to look at every single data point efficiently and effectively. So, statistics measures on the entire data set are used to derive intuition about the raw data set.

After learning the number of rows and columns in the data set, the names of the columns are printed, which immediately reveals that the dataset is unlabeled and dummy column names are used with format: var_{i}, where i is an integer from 0 to 199. Then the first five rows of data are peeked at, which further revealed the column is also a dummy column with data values in format: train_{i}, where i is the row number. Eyeballing other columns in this part provided the intuition that most values in this dataset are small floats. Next, with info() method, it was made certain that all columns are continuous variables, there are no missing values in the entire dataset, and all are floats. Now, with describe() method, the positional statistics and range of values for some of the columns could be assessed, which provided further intuition that the column values lied in approximately the same range. Histograms of all columns are plotted in a single frame for convenient comparison, which provided further evidence of the coherent ranges of the distributions and that they were approximately normally distributed, with no heavy tailed distributions. Later, more elaborate plots of the distributions are made with different package. This still confirmed normal distribution for most of the columns but some of the more "common exception" characteristics included nonsmooth peaks at the center and some sudden spikes towards the tails of the distributions. Some of the slightly right aligned distributions included columns var_5 , var_26 , var_37, var_53, var_134 and some of the slightly left heavy distributions included columns var_12 , var_33, var_44, var_59, var_73, var_81, var_109.

With this initial understanding, the next steps aimed at finding relationship between the columns. The correlation matrix was calculated to find out linear regression coefficient among the columns. The attempt to visualize this relationship by pair plotting all the columns (i.e. 200 x 200 figures) turned out to be computationally very expensive and so was aborted. The scatterplot between var_0 and var_1 gave a circular distribution, implying $r = 0$ (almost). So, traversing the correlation matrix revealed that only eight pairs of columns have $r > 0.009$ and all less than 0.010. This revealed no further information regarding associativity between the columns. Therefore, further attempts were made with polynomial regression. Polynomials of degrees from 2 to 4 are tried to fit the relationship among the columns, however, no values pair of columns with r-squared value between 0.05 and 1 are discovered. Thus, this attempt too failed to unearth any redundant columns.

Moreover, boxplots are made to further understand the distribution of the columns. While many columns contain few or no outliers, like the columns var_37 , var_41 , var_55; some columns, like var_0 , var_2 , var_6, var_10, var_16, var_17 etc. have significant number of outliers, explaining the tail spikes in the histograms. Therefore, it puts more weight on the proposition that these “extreme” values may represent some characteristics of the columns rather than some experimental error.

Feature Engineering:

The low variance filtering method is adopted to eliminate redundant columns. A column of 200, 000 rows with most of the values tightly clustered around the mean does not reveal enough useful information for further analysis. Therefore, the 59 columns with variance less than 5 is eliminated from the dataset.

Next, all the columns are transformed using the RobustScalar class. This method is picked because the initial dataset contains large number of outliers and this method is least sensitive to outliers.

Clustering Analysis:

Since the provided data is unlabeled, this problem is approached with unsupervised machine learning techniques. The KMeansClustering algorithm is used to group the data into clusters. This algorithm is computationally expensive because it starts with a random selection of centroids and then iteratively changes the centroids to minimize the average distance of the points from the centroids. The default limit of 300 iterations in the sklearn library implementation is used in this project. So , in order to find the suitable number of clusters, the KMeans algorithm is run with different values of k (all even numbers from 2 to 25 and 21) and the within-cluster-sum of squared (wss) value is recorded for each k. As the value of k is increased, the time required to run the algorithm also increased. So the range of trial values is broken to provide flexibility, the hop between the values could have been increased if there was still significant decrease in the value of the cost function. Next, the cost function values are plotted against the number of clusters. This scatterplot shows steep decrease of the cost function at the beginning, but it starts to decrease when k is around 12 and almost tapers off when k is greater than 20. Now, since a large value of k would also increase the chances of overfitting the model to training data, k =16 seems a reasonable breakpoint.

Dimension Reduction:

Next, the dataset is standardized by transforming with the StandardScaler class from sklearn library to meet the conditions for applying the Principal Component Analysis (PCA). Then, the PCA algorithm is run with different number of dimensions and their accuracy is logged. In PCA, the accuracy is measured by the quantity : average squared perpendicular distance from the point to their projection in lower dimension divided by the variance of the column. After running the PCA algorithm, it is discovered that 95% of the variance in the original data set is retained with 134 principal components/dimension, around 90% variance is retained with 126 dimensions, 85% is retained with 119 dimensions, almost 80% is retained with 112 dimensions and almost 70% is retained with 98 dimensions. Therefore, the dimensions of this dataset could not be reduced a great deal.

Results:

Exploratory data analysis phase revealed that the original dataset is clean, with no missing values and all columns contain continuous values of float type. It is also derived from positional statistics that all the data values are contained in a relatively small range, thus eliminating requirement for scaling due to data recorded different units. Histogram of each of the columns also revealed that most of the columns are approximately normally distributed, and none of them is drastically tail distributed. However, box plot of the distributions revealed many of the distributions contained significant number of outliers. The outliers are not handled because the presence of large number of such values could imply, they are true values and did not originate from random experimental errors, thus containing meaningful characteristic of the data. Then, the linear regression is carried out to uncover correlations among the features. However, no redundant column could be discovered in this method as well, with the highest correlation coefficient among the 200 x 200 pairs of columns being 0.09. Next, the low variance filter method is applied to eliminate columns with low variances. This analysis revealed that 59 of the columns have variance less than 5, so they are dropped from the dataset. With the first column containing only dummy values, the dataset is reduced to 141 columns. Then, K-Means clustering algorithm is applied to find patterns in this dataset. The K-Means algorithm is executed with different values of k and the relationship between the number of clusters and the error function is plotted. With the elbow method in action, k = 16 is taken as a reasonable estimate of the number of clusters. Finally, the Principal Component Analysis (PCA) algorithm is applied in effort to reduce the dimension of the dataset without losing any useful information. However, this result also indicates that 134 of the original 141 columns would have to be retained to preserve 95% of the variance of the data. Even preserving just 70% of the variance would require retaining 98 columns. Thus, the statistical analysis suggests the at least 130 out of the 200 columns should be retained to preserve more than 90% variance present in the original dataset.

One particularly interesting feature in the original dataset is the presence of significant number of outliers. In the case that the outliers are imputed with the mean of the column, the result of running the PCA algorithm might have been different, possibly capturing more variance with fewer dimensions of data. Also, carrying out hierarchical clustering might have suggested more about the differences in the data values and their structure. Moreover, having a little bit more information about the context of the provided data could have provided more room for analysis. Domain knowledge about the features represented in the columns could have allowed using statistical methods like putting the continuous values into discrete bins for further investigation. It could have also provided intuition for multiple regression; since it is also a possibility that the columns represented the same feature in different units, but the conversion did not change the range of the values. Furthermore, depending on the use case, different number of clusters could have been derived to suit the requirements and using the evaluation and test results from the machine learning models better estimate of the number of principal components could have been derived in the PCA analysis stage.