Fast Food Classification by Utilizing Principal Component Analysis and Machine Learning with PyCaret

Mohammed Abdullah - 40293614

GitHub: https://github.com/m8888e/INSE6220-PROJECT

***Abstract*—This report explores the use of Principal Component Analysis (PCA) and Machine Learning models to analyse nutritional data of several fast-food places. Using dimensionality reduction techniques, this study aims to reduce the computational burden for all involved parties, considering the fast-food industry's potential influence on consumer choices along with the ongoing public health concern. By recognizing dominant components (linear combinations), PCA helps mitigate redundancy in features and is beneficial for data visualization. The three classification algorithms (Logistic Regression, QDA, and Random Forest) are tested on both the original dataset and the dataset with PCA applied to it with PyCaret. Performances metrics showed that Logistic Regression always produced the highest accuracy and robustness. We used SHAP (Shapley Additive Explanations) to interpret contributions of principal components providing insights into feature importance and improving model transparency. The findings of this study highlight a valuable application of machine learning and explainable AI in nutrition data analysis to help individuals make informed decisions about their food choices and promote healthier eating habits.**

Keywords—Principle components analysis, Binary classification, Logistic Regression, Quadratic Discriminant analysis, Random Forest

I. Introduction

Fast food has firmly established itself as one of the most influential aspects of modern consumer culture, both embodying and encouraging major changes in diets and lifestyles. They have redefined food convenience to people who now want readymade, cheap and easily accessible foods that can pace up with the rapid lives of a large proportion of consumers today. Due to this, fast food has affected the way people prepare their meals by raising the number of absences of homes for hot meal a day by replacing quick and simple home cooked dishes with solutions which are simply heated [1][2]

The fast-food industry has also affected health and eating habits by narrowing consumer tastes towards calorie-dense, highly processed foods. This has stoked public health fears, reopening debates about nutrition in schools and issues such as obesity and wellness. Thus, fast food is not just the solution to consumer demands for convenience but also a symbol of wide-ranging changes in our lifestyle, eating styles and the perception towards food at large in modern-day capitalism-driven societies [2].

Within this fast-food dataset, we can analyse the nutritional makeup of common items and get a perspective on the diet of many people today. By allowing for protein, fat and carbon units in the balance we are able to assess the trade-off between health goals and marketing. Junk content analysis also yields information regarding the mineral composition of these products, which may serve to inform consumer choice on health grounds. The in-depth analysis of sodium, which is zan important but hotly contested nutritional phenomenon of fast food that bears consideration toward its diversity between categories and public health consequences, takes it a step farther [1].

Techniques for exploring nutritional data in fast-food environment are complex and interdependent requiring more advanced techniques to analyse the data. Dimensionality reduction techniques such as Principal Component Analysis (PCA) are especially useful in this case. This allows us to represent many nutritional metrics in terms of orthogonal components, making it easier to see patterns that might relate some attributes and not others when PCA is applied on the fast-food industry dataset with common nutrition facts. Such a simplification of the dataset for analysis will not only assist researchers in determining dominant factors that could play a role in consumer choice and health outcomes but can also help the sector move into offering healthier balanced offerings, guided by more consumer choices.

II. Principle Component Anaysis

Almost all of the datasets have high dimensionality. The processing and storage of these types of datasets is very costly, in some cases even visualizing the resultant data would be impossible. PCA can be a great help in this case, it is featuring reduction technique that helps you to reduce the number of features from a large data set by being able to transform a large set of variables into a smaller one while retaining most of the information. PCA is a great method to simplify high dimensional data whilst keeping the trends and patterns. It achieves this by storing the data in lower-dimensional spaces that act as a feature summary [3].

A. The PCA Algorithm

PCA can be performed on a data matrix X of dimension (n × p) via the following well-defined four steps [4]:

For step 1: Standardization, it is the first and the most important thing that you need to do in order for data analysis to be effective as it enables each variable work equally towards the results. The mean vectors $\bar{x}$ for each of the

columns in the dataset are calculated first. The mean vector, a *p* dimensional summary of our data, is written therefore as [4]:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Each value in the data matrix is converted by subtracting the column mean. The matrix of the cantered data (Y) can be represented by the following [4]:

$$Y = HX$$

Then, for step 2: it is important to calculate the covariance matrix (p × p) from this centered data in order to find the corelations and it is computed as follow [4]:

$$S = \frac{1}{n-1} Y^T Y.$$

For step 3, for the covariance matrix S, the eigen values and vector of S can be computed using the eigen decomposition. The directions of each principal component (PC) are indicated by eigenvectors while the amount of variance captured by each PC is represented in eigenvalues so it can be computed as [4]:

$$s = A\Lambda A' = \sum_{i=1}^{p} \lambda_j a'_j a_j.$$

A is an p x p orthogonal matrix (i.e. A'A = I) with columns aj = (aj1; aj2….. aj_p), which are the eigenvectors of S. Λ=diag (λ1;λ2….λp) is a p x p diagonal matrix containing the eigen values of S in descending order.

For step 4, n this fashion, we compute the transformed matrix of size n × p Z where n is the number of observations and p represents the PCs [4]. The number of PCs is equal to the dimensions of the original data matrix. So, the equation is Z=Y*A which has dimension (nxp)

$$Z = (z'_1, z'_2, ..., z'_i, ..., z'_p) = \begin{bmatrix} z_{11} & z_{12} & .. & .. & z_{1p} \\ z_{21} & z_{22} & .. & .. & z_{2p} \\ . & . & . & . & . \\ . & . & . & . & . \\ z_{n1} & z_{n2} & .. & .. & z_{np} \end{bmatrix}$$

The transformed data matrix has the original samples represented in the new coordinate system defined by Principal Components. Observations (Zi = A'(xi-$\bar{x}$)) are the rows of Z matrix while columns correspond to PC scores

B. Random Forest Classifier

Random Forest is a very popular and effective classification or regression machine learning algorithm. It is a type of ensemble method, which means that it trains several different individual models which combined makes more accurate and robust prediction. To be more specific, a Random Forest is a collection of decision trees that is trained on a random sample of the data using bootstrap sampling. Compared to each of the splits in a tree, only a subset of the features is taken from a random selection, to keep the trees diverse. The core idea of a Random Forest is to combine the predictions made by each of its constituent individual decision trees, aggregating those predictions using either majority voting (in the case of classification problems) or averaging (in regression problems) which allows the ensemble to reduce overfitting which is one of the main weaknesses of single decision trees [5].

C. Logestic Regression Classifier

Logistic Regression is a statistical method for binary classification problems in which the outcome or dependent variable is categorical and has two possible outcomes (e.g., yes/no, 0/1, true/false). It is a common approach in machine learning and statistics because it estimates the likelihood of a given input being part of some category. It uses the sigmoid function to map predicted values to probabilities [6]:

$$S(z) = \frac{1}{1+e^{-z}}$$

Logistic Regression LR is a simple, efficient and widely used classification algorithm. For small to medium size data sets it does well overall and gives some indications of what features are driving the results, also as a probability output it allows you to vary the threshold on predicting a fraudulent account. It can deal with both continuous and categorical variables, as well as include L1 and L2 regularization methods to address overfitting. It is resilient to noise and outliers, but performs best on linearly separable data, and may fail for non-linear relationships [6].

D. Quadratic Discrimeinant Analysis QDA

QDA is a classification algorithm similar to LDA (Linear Discriminant Analysis) but it assumes that the decision boundary between classes can be quadratic. It simplifies the assumption about covariance matrix by being able to allow differ variance across classes. Also, it identifies class conditional densities and applies Bayes theorem to classify new data points in the maximum posterior probability class. QDA assumes that the class conditional densities can be estimated as Gaussian distributions and then holds Bayes' rule as follow [7].

$$f_c(x) = \frac{1}{\sqrt{(2\pi)^n |\sum c|}} e^{-\frac{1}{2}(x-\mu_c)^T \sum_c^{-1} (x-\mu_c)}$$

Where x: input features, μ*c* is the mean vector and Σc is the covariance matrix of class c, while n is equal to the number of features.

III. DATASET DECRIPTION

The dataset was retrieved from Kaggle and the dataset has over 900 samples and 8 features (including class) for each sample. Values in the Region column appear to be whole numbers that are likely identifying specific regions. The columns MajorLength and MinorLength are two floating-point measurements of the major and minor lengths of the samples, respectively. The Elongation column also represents a floating point value — that is, the ratio of elongation. The

Convex Region column is an integer value which may represent the convex region of each sample. Moreover, the Spread column logs a decimal representation of the spread for that sample, and Boundary length saves a double value for the total length of boundary for that sample. Finally, the class column represents a binary classification label with values of either 0 or 1.

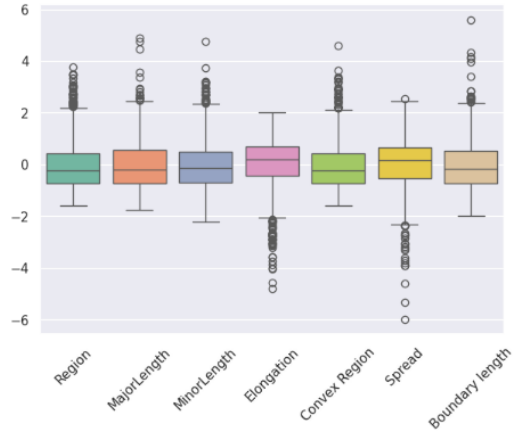

Figure 1: boxplot

As shown in figure 1, The boxplot shows eight numerical variables for multiple categories probably across regions or groups. One common visualization used for comparing numerical data distributions is the boxplot. Most of the features are approximately normal. On the contrary, there are outliers in all features. Except for Elongation features, all features have left outlier while Spread has outlier on both sides.

IV. PCA Result

PCA on fast-food dataset. PCA has two approaches to implement, the first is from scratch using generic Python libraries e.g. NumPy and the second is a proper documented and popular PCA library via wrapper. Kaggle notebook demonstrates both the methodologies. The results from both approaches are similar but the PCA library is more flexible, allowing users to complete tasks in just one line of code. The numbers and plots for this report are based upon the running of PCA library.

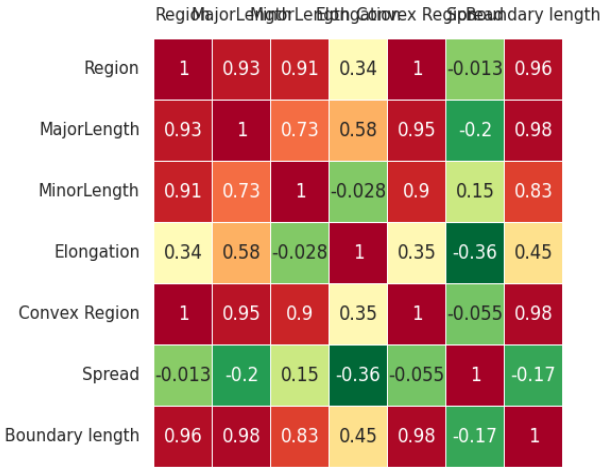

Figure 2: Correlation matrix

Figure 2 illustrates visual representation of the interrelationships among a set of numerical variables. The variables are Region, Major Length, Minor length, Elongation, Convex area, Spread and Boundary Length for this matrix. The correlation matrix: Each cell of a matrix gives the correlation between two variables, each variable is represented by a row and a column. The coefficients are between -1 and 1 The value 1 means there is a perfect positive correlation, meaning as one variable increases the other also increases in direct proportion. On the other hand, -1 means a perfect negative correlation: when one variable increases, the second decreases. A score of 0 means that the variables are not correlated to each other.

From the matrix there are some patterns that can be picked from it. As an example, "Region" is highly positively correlated with "Major Length"(0.93) and "Boundary (0.96). This means that as regions become larger, they more often have major dimensions and boundary lengths which are longer. The same applies for "Major Length" and "Minor Length", which have a moderately strong positive correlation of 0.73, meaning if the major axis is longer then often times the minor one will be too but not necessarily. Such connections imply that areas have interrelations in size and form.

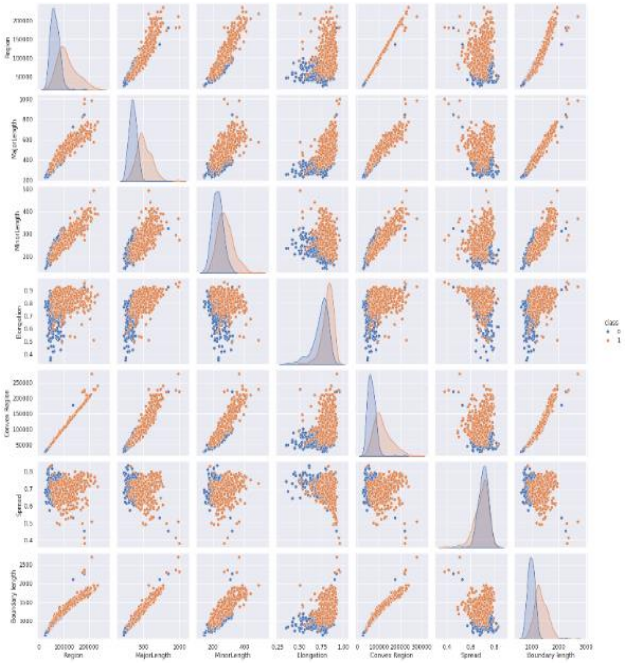

Figure 3: pair plot

The pair plot gives us insights into how features relate to each other and their distribution in the dataset. Scatterplots between feature pairs show strong linear relationships in some instances (such as between Region and Convex Region and between Region and Boundary Length). The relationship suggests a high redundancy among these features because they seem to capture closely related traits. Also, with MajorLength and Boundary Length the linear trend is clear, meaning one tends to grow when the other grows too; they influence each other. Meanwhile, features like Spread do not share relatively clear patterns with others which indicate its independence that makes it a strong predictor on its own when used in a ML model

```
Eigenvector matrix:
[[ 0.45599636 -0.0955827   0.02984836  0.08621358  0.53411075 -0.19544386
   0.67166581]
 [ 0.44241095  0.16847249  0.14989626 -0.56894    -0.17101774 -0.58393244
  -0.243934  ]
 [ 0.37970747 -0.42439546 -0.22282621  0.58266084 -0.4397083  -0.28257888
  -0.11563451]
 [ 0.15605343  0.69041146  0.48811898  0.49769217 -0.11380621 -0.00686512
  -0.00476786]
 [ 0.45682821 -0.0849584   0.01203094  0.0768239   0.50284372  0.36428046
  -0.62649129]
 [-0.09597179 -0.54599352  0.82978933 -0.03583781 -0.04343813  0.0309406
  -0.00127291]
 [ 0.4581559   0.01935533  0.00578392 -0.27294155 -0.47375597  0.63817636
   0.28891904]]
```

Figure 4: Eigenvector matrix

From the matrix, first two Principal Components are:

$Z1$=0.455 X_1 +0.442 X_2 +0.379 X_3 +0.156 X_4 +0.456 X_5 -0.095X_6+0.458X_7

As we can see from first PC, x1, x5 and x7 contribute to be the highest contribution but none of the features have a negligible contribution.

$Z2$=-0.095 X_1 +0.164 X_2 -0.424 X_3 +0.690 X_4 -0.084 X_5 -0.545X_6+0.0193X_7

As we can see from second PC, X_4, X_5 contribute to be the highest contribution and X_1, X_7 and X_5 can be negligible. So we rewrite:

$Z2$=0.164X_2-0.424X_3+0.690X_4-0.545X_6

Also, the Eigen Value are: (4.7940, 1.4731, 0.7749, 0.0531, 0.0183, 0.0049, 0.0004)

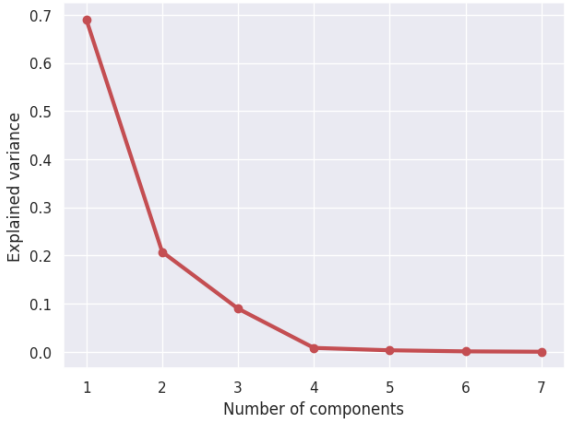

Figure 5: Scree plot

Figure 5 depicts plot of the variance explained by each additional principal component PC in terms of percentage. The x-axis is the number of principal components and y proportion of variance explained by individual component. The major drop in explained variance from first to second component means almost all the dataset variance is contained in its first principal component ~70% of total variability and the elbow shape r = 2. The second component explains much less variance ~20%, but still provides substantial information. Starting from the third component, the explained variance starts to decline even further and every subsequent component adds just a tiny portion of total variance. The proportion of explained variance reaches is maximum for the fourth component, after which it plateaus, indicating that these components add only negligible information.

Moreover, figure 6, show Pareto chart which shows the first two components explain most of the variance as they are responsible of about 90 % total variability in the data (L1=69% for PC1 and L2=20.7% for PC2). The change in cumulative explained variance plateaus after the second component, as subsequent components add little to the total explained variance.

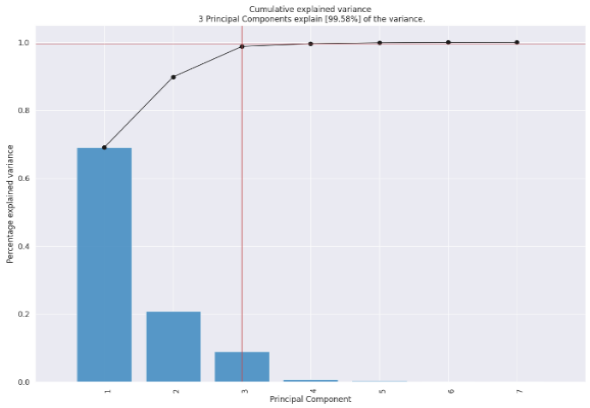

Figure 6: Pareto plot with explained variance

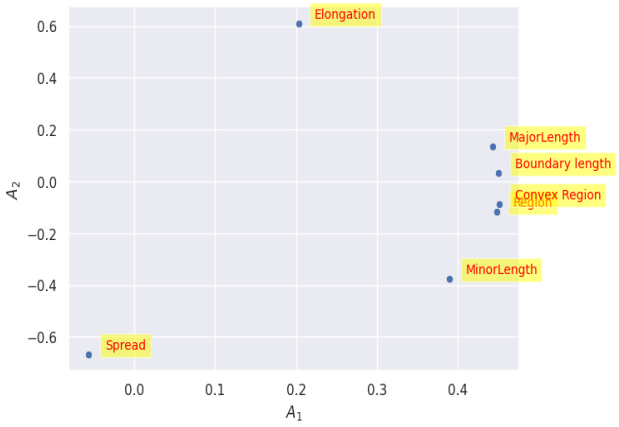

Figure 7: PC coefficient plot

As shown in figure 7, the coefficient plot focusses on the loadings of features within the first two PCs, corroborating findings with the biplot. PC1 is predominantly determined by features such as MajorLength, Boundary Length and Convex Region indicating that this component accounts for variability in geometric measures. On the other hand, Elongation exhibits highest loading on PC2, which captures orthogonal information. The combined effect of Spread tells a different – and weaker yet conflicting – story on the dataset, which means that we might have more variability (high level uniqueness with limited contribution) from spread.

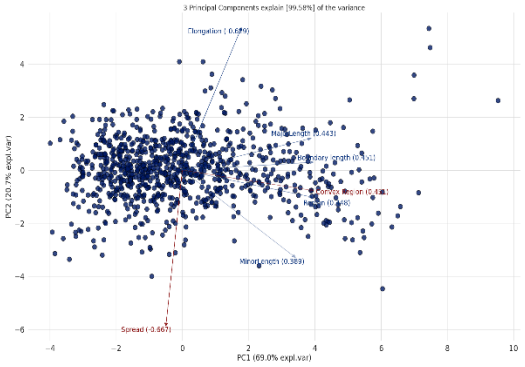

Figure 8: Biplot

The PCA biplot in figure 8 shows the projections such that observations and feature vectors are projected onto the PC1 and PC2 axes. Notice that in the plot above, the data points can be assigned to "Label 0" and "Label 1" classes but are mixed together, meaning that although information for class separation is not completely lost after projecting onto the first two PCs, there is still partial overlap. Some feature vectors, such as Elongation, are closely aligned with PC2, indicating that this feature contributes significantly to variability along the vertical axis, while Spread is closely and negatively aligned with PC1 (an inverse contribution). PC1 is primarily driven not only by MajorLength and Boundary Length, which mainly applies to horizontal, but also indicates that they are related to ones.

V. CLASSIFICATION RESULTS

In this section, three well-known classification algorithms are performed on the fast-food dataset for comparison purpose. The aim is to examine how PCA will affect the provided data. In order to reach this goal, the classification algorithms are applied in two kinds of dataset: original and PCA-transformed dataset which is obtained after three principal components. PyCaret was leveraged to produce a performance comparison table that has all important classification algorithms for the home data set. Through these tables, we can understand which algorithm provides the highest classification accuracy and help us to select the best model.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8677	0.9311	0.8432	0.8862	0.8628	0.7351	0.7380	0.4330
qda	Quadratic Discriminant Analysis	0.8659	0.9242	0.7898	0.9329	0.8522	0.7314	0.7434	0.0400
ridge	Ridge Classifier	0.8625	0.9317	0.8469	0.8740	0.8588	0.7248	0.7274	0.0280
rf	Random Forest Classifier	0.8625	0.9300	0.8151	0.9011	0.8534	0.7247	0.7311	0.3340
et	Extra Trees Classifier	0.8572	0.9294	0.8148	0.8922	0.8483	0.7140	0.7210	0.1500
lightgbm	Light Gradient Boosting Machine	0.8572	0.9246	0.8366	0.8710	0.8523	0.7144	0.7165	0.3170
lda	Linear Discriminant Analysis	0.8554	0.9318	0.8505	0.8593	0.8533	0.7106	0.7132	0.0250
xgboost	Extreme Gradient Boosting	0.8519	0.9158	0.8261	0.8722	0.8455	0.7037	0.7088	0.0890
ada	Ada Boost Classifier	0.8501	0.9179	0.8151	0.8782	0.8424	0.7001	0.7059	0.1560
gbc	Gradient Boosting Classifier	0.8483	0.9240	0.8079	0.8805	0.8392	0.6963	0.7029	0.1800
knn	K Neighbors Classifier	0.8255	0.8912	0.8009	0.8410	0.8193	0.6507	0.6529	0.0400
nb	Naive Bayes	0.8253	0.9150	0.7475	0.8819	0.8078	0.6501	0.6587	0.0260
dt	Decision Tree Classifier	0.8131	0.8128	0.8081	0.8241	0.8105	0.6259	0.6340	0.0290
svm	SVM - Linear Kernel	0.5806	0.8312	0.8643	0.5335	0.6405	0.1647	0.1753	0.0270
dummy	Dummy Classifier	0.5044	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0230

Figure9: Classifications comparison before applying PCA

As shown in table in Figure 9, From the table above, Logistic Regression is the best classifier among all. With a highest accuracy of 86.77% means that it has classified the most of the test data correctly as compared to other models. Moreover, not only is Logistic Regression accurate but also shows comparable performance across other metrics: Precision (88.6%), F1-score (86.2%), Kappa (73.5%), and MCC (74.8%). The model has fairly good predictions as proposed by these values suggesting that it predicts quite well indeed and does not learn at the expense of precision or recall.

Additionally, the time efficiency that results from Logistic Regression contributes to its being chosen as the best classifier one. It operates with execution time 0.4330s, and is both effective and computably feasible for actual use. Some models such as Linear Discriminant Analysis (0.93.1% AUC), Quadratic Discriminant Analysis (highest precision metric of 93.29%) outperform it on certain metrics but are not able to keep the consistency as Logistic Regression, across all metrics.

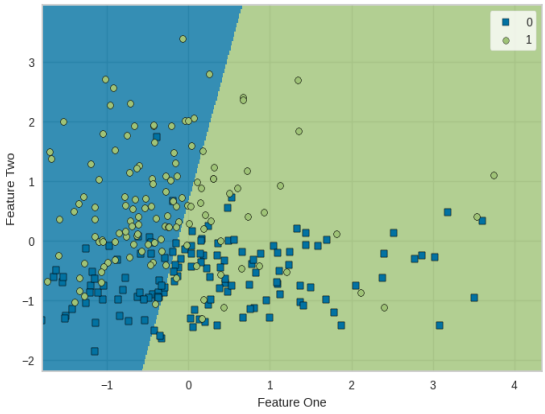

Figure 10: Logistic Regression decision boundary

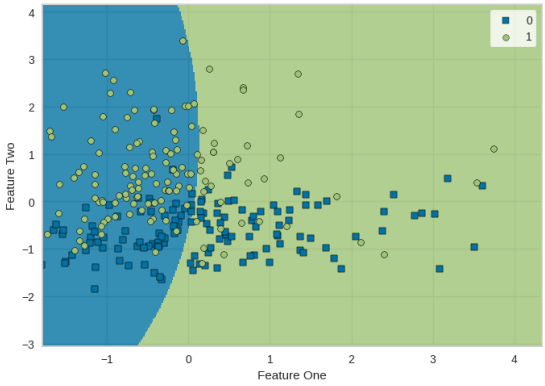

Figure 11: QDA decision boundary

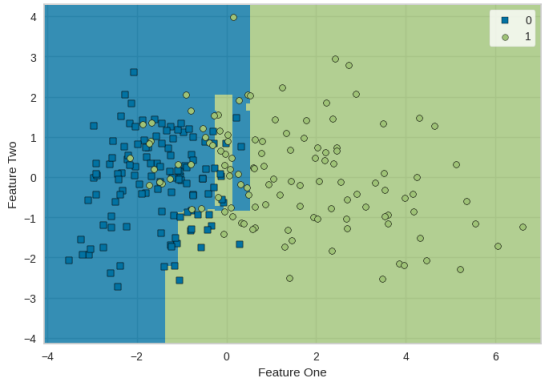

Figure 12: Random Forest Decision boundary

Form figure 10, 11 and 12. The three classifiers, the Logistic Regression, Quadratic Discriminant Analysis (QDA), and Random Forest—have quite different decision boundaries. Logistic Regression gives us a simple linear boundary cleanly separating the classes with a straight line. While conveying instructions intuitively, this hyperplane only accurately predicts linearly separable data and can potentially fail on non-linear data such that overlapping sections are still unable to correctly classify near the hyperplane.

Meanwhile, with QDA, a curved decision boundary is constructed providing more flexibility to accommodate non-linearities between the features. Unlike LDA, which assumes the same covariance for each class, QDA (quadratic discriminant analysis) relaxes this assumption and allows each class to have different covariances, resulting in more complex separation and smooth transitions between classes. Assimilation of the only statistical properties of the class

distributions works out well if the data does not have linear distribution.

Since Random Forest is used to build multiple decision trees and builds an ensemble, the decision boundary of Random Forest is a complex and irregular boundary. This allows it to learn complex and nonlinear relationships, which is very useful for modeling, liberating the algorithm. But the sharp rectangular areas that result may suggest a risk of overfitting, particularly in noisy data.

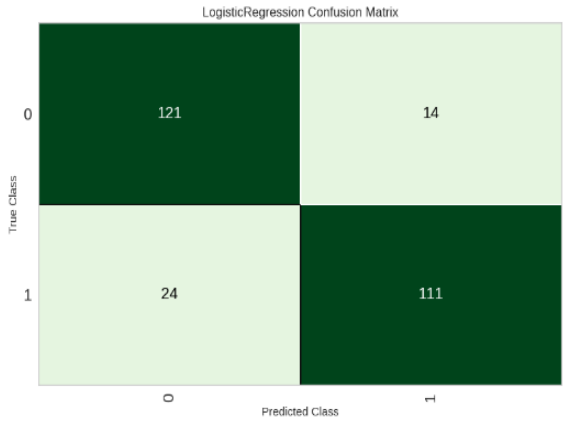

Figure 13: Logistic Regression LR confusion matrix

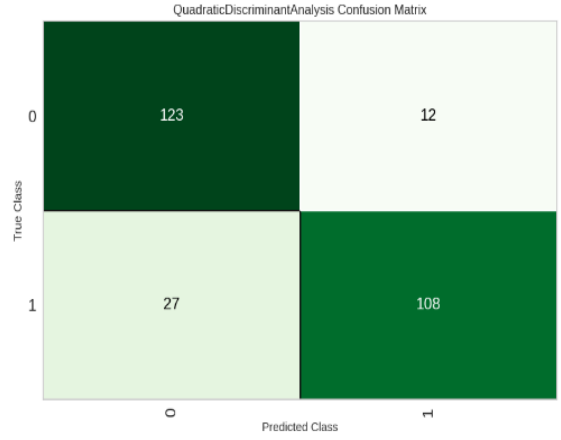

Figure 14: QDA Confusion matrix

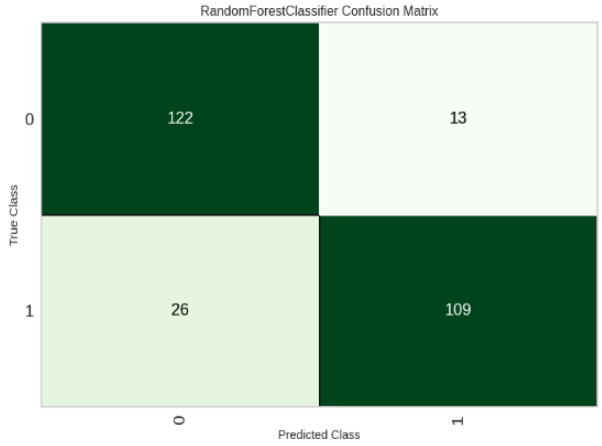

Figure 15: Random Forest Confusion matrix

From figures 13, 14 and 15 show Logistic Regression LR, QDA and Random Forest models, the LR is the best-balanced algorithm with the lowest false negatives which here is 24 indicating that it correctly identifies Class 1 samples well while being good overall. Quadratic Discriminant Analysis (QDA) shines among the other classifiers and achieves the lowest false positives (12), indicating good performance towards Class 0 being non-overlapping. Its false negatives (27) are a little higher than those of Logistic Regression, though.

The Random Forest Classifier produces a weighty performance for the overall classifications of all classes in the dataset hence it proves to be a strong balanced classifier in the task. So, it has correctly predicted a 122 true negatives (class 0) and 109 true positives (class 1), which means it can handle both the above categories very well. But it yields 13 false positives (class 0 classified as class 1 incorrectly) and 26 false negatives (class 1 classified as class 0 incorrectly). Although it yields a slightly higher count of false negatives than the Logistic Regression model - providing a readout of a lower false positive rate - which shows that Random Forest is more robust against alarmist predictions of class1.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8619	0.9310	0.8510	0.8715	0.8595	0.7240	0.7265	0.0330
ridge	Ridge Classifier	0.8587	0.9305	0.7972	0.9099	0.8469	0.7177	0.7259	0.0600
lda	Linear Discriminant Analysis	0.8587	0.9305	0.7972	0.9099	0.8469	0.7177	0.7259	0.0320
gbc	Gradient Boosting Classifier	0.8540	0.9222	0.8290	0.8741	0.8490	0.7082	0.7118	0.1450
rf	Random Forest Classifier	0.8508	0.9181	0.8258	0.8704	0.8462	0.7018	0.7046	0.2420
ada	Ada Boost Classifier	0.8492	0.9117	0.8098	0.8797	0.8414	0.6986	0.7031	0.1170
svm	SVM - Linear Kernel	0.8476	0.9216	0.8321	0.8679	0.8420	0.6954	0.7049	0.0510
nb	Naive Bayes	0.8460	0.9274	0.7623	0.9165	0.8294	0.6923	0.7045	0.0490
xgboost	Extreme Gradient Boosting	0.8460	0.9013	0.8286	0.8598	0.8429	0.6922	0.6942	0.0690
knn	K Neighbors Classifier	0.8429	0.9083	0.8100	0.8676	0.8356	0.6860	0.6900	0.0470
et	Extra Trees Classifier	0.8429	0.9217	0.8259	0.8587	0.8402	0.6860	0.6893	0.1570
qda	Quadratic Discriminant Analysis	0.8302	0.9157	0.7497	0.8954	0.8130	0.6605	0.6722	0.0310
lightgbm	Light Gradient Boosting Machine	0.8286	0.9019	0.8100	0.8435	0.8248	0.6574	0.6603	0.1800
dt	Decision Tree Classifier	0.8048	0.8049	0.7875	0.8179	0.8011	0.6096	0.6120	0.0560
dummy	Dummy Classifier	0.4921	0.5000	0.5000	0.2460	0.3298	0.0000	0.0000	0.0500

Figure 16: Classifications comparison after applying PCA

In figure 16, after applying PCA, Logistic Regression LR is the best model with an accuracy of 0.8619 and F1-score (0.8595), continuing to trump all else on Kappa(0.7240), signifying high consistency and efficiency. Naive Bayes has the highest precision but does not perform well in terms of gaining overall accuracy and generalization. However, other models including Ridge Classifier and Linear Discriminant Analysis (LDA) are close to this accuracy with 85.87% accuracy, they are lower than Logistic Regression. Their precision and F1-scores do not vary much with each additional vectorization, which indicates that linear models remain suitable after dimensionality reduction. QDA (Quadratic Discriminant Analysis) is a non-linear model with a performance of ~2% worse than above, at 83.02%, likely because PCA eliminated some of the features that contributed to non-linear decision-making.

```
# Tune hyperparameters with scikit-learn (default)
tuned_best_model_pca = tune_model(best_model_pca)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8730	0.9556	0.8710	0.8710	0.8710	0.7460	0.7460
1	0.8571	0.9244	0.9032	0.8235	0.8615	0.7146	0.7179
2	0.8254	0.9093	0.8065	0.8333	0.8197	0.6505	0.6509
3	0.8730	0.9123	0.8710	0.8710	0.8710	0.7460	0.7460
4	0.8730	0.9506	0.8387	0.8966	0.8667	0.7457	0.7472
5	0.8254	0.8881	0.7188	0.9200	0.8070	0.6519	0.6685
6	0.8413	0.9194	0.8125	0.8667	0.8387	0.6828	0.6842
7	0.8571	0.9325	0.8438	0.8710	0.8571	0.7144	0.7147
8	0.8571	0.9315	0.8125	0.8966	0.8525	0.7146	0.7179
9	0.9683	0.9970	0.9688	0.9688	0.9688	0.9365	0.9365
Mean	0.8651	0.9321	0.8447	0.8818	0.8614	0.7303	0.7330
Std	0.0384	0.0286	0.0630	0.0397	0.0413	0.0766	0.0749

Figure 17: Logistic Regression LR metric score after hypermeter tuning

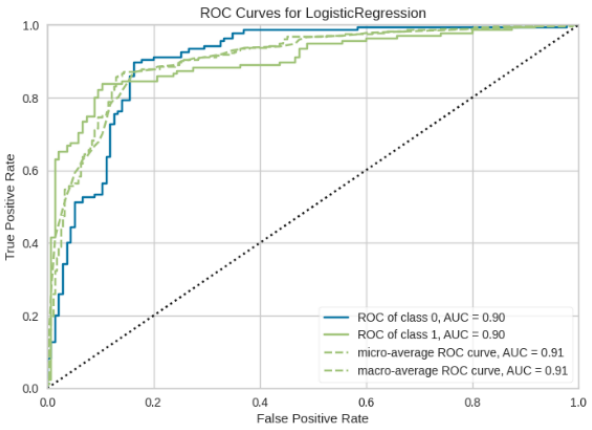

Figure 18: ROC curve for Logistic Regression LR

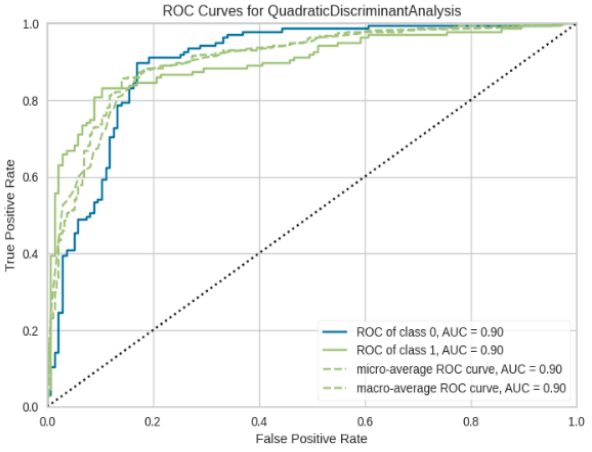

Figure 19: ROC curve for QDA

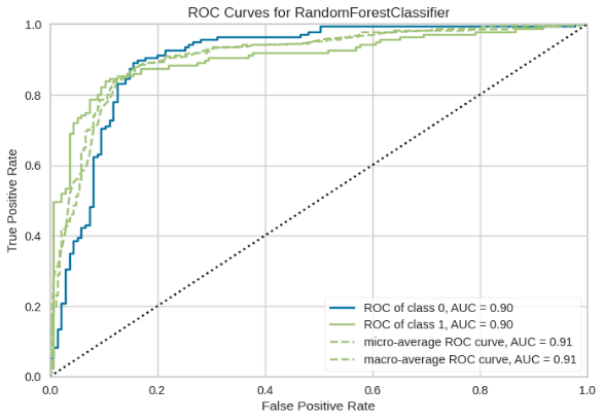

Figure 20: ROC curve for Random Forest

The ROC curve for Logistic Regression in figure 18 shows very good performance;.90 AUC for Class 0 and Class 1, with macro-average AUC equal to having a point of 0.91. Both classes have still a close curve to top-left corner for high True Positive Rate (TPR) and low False Positive Rate (FPR). Indicating that Logistic Regression and it is able to separate the classes properly, achieving a relevant balance between precision and recall. As the linear model, its simplicity enables it to get consistent results.

The QDA model has comparable performance with AUC of 0.90 for the AUC and 0.90 for the macro average AUC as shown in figure 19. There is relatively little difference between this and the ROC curves for Logistic Regression, except in the lower FPR range where TPR increases gradually. This is an example of the flexibility of QDA to model non-linear decision boundaries, giving it an edge on more complex data distributions. But it still performs somewhat worse than Logistic Regression and Random Forest regarding optimal early separation.

In figure 20, The Random Forest Classifier can be seen giving a great performance with an AUC of 0.90 for both classes and a macro-average AUC of 0.91. The Random Forest ROC Curve is smoother and demonstrates a higher TPR at lower FPR compared to Logistic Regression and QDA. It indicates that Random Forest is capable in separating classes, especially for a complicated data set, as it has the ability to catch non-linear relationships and interactions between attributes.

VI. EXPLAINABLE AI WITH SHAPLY VALUES

Explainable AI (XAI) describes various techniques and methods that render machine learning models more transparent and interpretable. XAI plays an essential role in being able to see why a model made a prediction, which can lead to trust, aid in decision making, and enable accountability of the model. SHAP (Shapley Additive exPlanations) is one of the most powerful methods for explainability, based on cooperative game theory concepts (specifically Shapley values). SHAP values explain each predictive feature's contribution in a fair way, providing an explanation that is consistent and reliable [8].

SHAP Values explain how a Machine Learning model is making a prediction by providing the attribution of the prediction to each feature. For a particular instance, SHAP

computes how much does each feature contributed to the prediction outcome by comparing it with what would be the outcome if the feature was not there. With SHAP, all possible combinations of input features are accounted for, allowing for more equitable distribution of the contribution from each feature effortlessly [9]. However, since SHAP values consider both background and actual inputs, it proves to be an effective explanation for both single predictions (local interpretability) and also as we see the total score for each feature over the dataset (global interpretability).

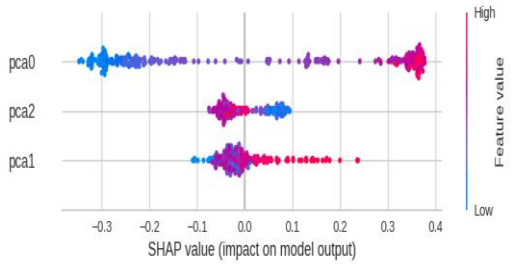

Figure 21: Summary Plot

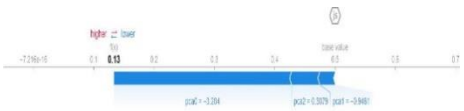

Figure 22: Force plot

Figure 21 illustrates SHAP summary plot that shows how the three principal components (pca0, pca2, and pca1) influence the machine learning model's prediction. SHAP values, are plotted on the x-axis for each feature — showing the impact of each feature on the prediction made by the model. The feature values are plotted on y-axis, while the range is shown with color (blue low, red high). According to the plot above, pca1 has the largest impact since higher values tend to correspond to higher values of outputs of the model. pca2 also has a significant but lesser effect than pca1. Also, compare to that, pca0 has very low impact to the model prediction. The distribution of SHAP values for each feature indicates how much its effect varies across different points.

The prediction for this instance comes out to be 0.13, which is considerably lower than the base value. This decrease is mainly due to the contributions of the features shown as blue bars, which make a negative contribution to the prediction. The biggest winner is pca0, at -3.204, so it pulls the prediction quite a bit down. The other feature, pca1, also negatively contributes as well but not as much as pca0 (-0.9481). On the other hand, pca2 contributes a non-negative (0.3079) towards the prediction, but not enough to outweigh the offensive impact of the other two variables.

The direction and magnitude of the contributions are essential to interpreting the behavior of the model. The blue bars display features that reduce the prediction; red bars (not shown in this plot) would correspond to features that raise the prediction. In this case every relevant feature is working towards lowering the value, guiding it down from its baseline of 0.5 in the prediction to the final value of 0.13. This shows that some of the features in the input data have a much stronger negative impact than others on the output.

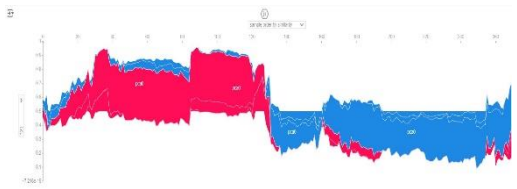

Figure 23: SHAP values contribution across all predictions

As shown in figure 23, the SHAP area chart tells us that the factor pca0 is having the biggest influence of all on the predictions from the model. Its contributions vary, at times boosting predictions (positive SHAP values in red) and at times depressing them (negative SHAP values in blue). Prediction per sample as generated by environmental data; accounts for positive and negative per sample, indicates a region of pca0 where prediction is lower and higher within the scatter. This implies that, pca0 has a high non-linear relation to the model output

VII. CONCLUSION

All in all, this study shows that PCA is an excellent approach to the dimensionality reduction of the datasets but nevertheless highly retains important information, given that the first two components account for almost 90% of the variance of the transformed dataset. Of all the classification models used, Logistic Regression performed better than the rest in accuracy, precision, and score stability, be it before or after applying PCA. Random Forest and QDA also performed moderately well, but both displayed issues with consistent and inefficient prediction. Using SHAP values, the study further identified feature importance, noting that pca0 had the greatest impact on predictions. In conclusion, this research addresses the effectiveness of machine learning and explainable AI in extracting insights from fast-food data to promote more healthy food choices, resulting in better choices made as to what can be consumed.

REFERENCES

[1] Wahlqvist, M. L. (1996). Food habits in later life: A Cross Cultural Study. Food and Nutrition Bulletin, *17(4), 1–1*. https://doi.org/10.1177/156482659601700434

[2] Fuhrman, J. (2018). The hidden dangers of fast and processed food. *American Journal of Lifestyle Medicine*, *12*(5), 375–381. https://doi.org/10.1177/1559827618766483

[3] H. Abdi and L. J. Williams. (2010). "Principal component analysis," Wiley interdisciplinary reviews: computational statistics, *vol. 2, no. 4, pp. 433–459,*

[4] A. B. Hamza, Advanced Statistical Approaches to Quality. Unpublished.

[5] pro.arcgis.com. (n.d.). *Train Random Trees Classifier (Spatial Analyst)—ArcGIS Pro | Documentation*. [online] Available at: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/train-random-trees-classifier.htm.

[6] IBM (2021). *Logistic Regression*. [online] Ibm.com. Available at: https://www.ibm.com/think/topics/logistic-regression.

[7] Data Science. (2018). *Linear, Quadratic, and Regularized Discriminant Analysis*. [online] Available at: https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/.

[8] Arxiv.org. (2018). *A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME*. [online] Available at: https://arxiv.org/html/2305.02012v3.

[9] C3 AI. (n.d.). *Shapley Values*. [online] Available at: https://c3.ai/glossary/data-science/shapley-values/.