Classifying Product from their Ingredients Based on various Machine Learning Algorithms

Aharnish Pithva
AU2040022
Ahmedabad University
aharnish.p@ahduni.edu.in

Jevin Jivani
AU2040051
Ahmedabad University
jevin.j@ahduni.edu.in

Astha Bhalodiya
AU2040067
Ahmedabad University
astha.b@ahduni.edu.in

Yug Patel
AU2040181
Ahmedabad University
yug.p2@ahduni.edu.in

***Abstract*—The purpose of this report is to investigate the application of machine learning algorithms in classifying ingredients into two distinct categories: food or beauty product. Optical Character Recognition (OCR) technology is utilized to convert images of ingredient labels into text, which is then processed by a range of algorithms, including Decision Trees, k-Nearest Neighbors, Random Forest, Naive Base, Linear Regression, Logistic Regression and Support Vector Machines. The performance of each algorithm is thoroughly evaluated and compared using evaluation metrics such as ROC curve and Precision Recall Curve. Additionally, Principal Component Analysis (PCA) is implemented on the data, and GridSearchCV is employed to optimize the accuracy of the model. Furthermore, cosine similarity is applied on the dataset to recommend other similar products based on a given product.**

***Index Terms*—OCR, Classification, Logistic Regression, Linear Regression, k-Nearest Neighbors, Random Forest, Decision Trees, Naive Base, Support Vector Machine, PCA, GridSearchCV, Cosine Similarity**

I. Introduction

Product classification is an important task in many industries, including food and beauty. Classifying products based on their ingredients can be a complex process, but with the help of machine learning algorithms, it can be done accurately and efficiently. In this report, we explore the use of various machine learning algorithms for classifying products based on their ingredients. The ability to accurately classify products from their ingredients has numerous real-life applications. For example, it can be used to identify allergens in food products, which is crucial for individuals with food allergies or intolerances. It can also be used to identify harmful chemicals in beauty products, helping consumers avoid adverse reactions or skin irritations. Furthermore, the precise classification of products can be beneficial in developing a recommendation model that provides a list of similar items based on the given product. In our product, user can scan ingredient list from the product they have, and with help of OCR with get text data from photo and will use this list into our trained model and classify accordingly.

In this report, we explored various machine learning algorithms, including Decision Trees, k-Nearest Neighbors, Random Forest, Naive Base, and Support Vector Machines, to classify products based on their ingredients. We then compared the performance of these algorithms and evaluated their accuracy and efficiency in product classification. By the end of this report, we aim to provide insights into the use of machine learning algorithms for product classification based on their ingredients, and their potential applications in various industries.

II. Background/Literature Review

Initially, our project aimed to use OCR to extract text from images and will use NLP to comprehend the context to enable users to ask relevant questions about the image. This technology had potential applications in education, accessibility, and information retrieval. However, we realized that our primary focus was on the deep learning aspect, so we decided to modify our problem statement.

Our new problem statement centers around utilizing an existing OCR scanning model to extract text from images, which will then be fed into our classification model. The primary focus of this project is the classification of ingredients. Additionally, we aim to use a text data set collected from the output of our own images processed by the OCR model and data generated by using the API key of ChatGPT.

III. Implementation

A. Data Collection

We attempted to find data from various websites such as Kaggle and Snap, but were unable to find suitable Datasets. We have found some data of ingredients of US market but in our case, we are classifying indian products based on their ingredients, and we wanted to collect data specific to the Indian market. As a result, we decided to create your own dataset by manually collecting ingredient information and also generating more data using API key of ChatGPT for the products available in the Indian market.

B. Data Cleaning

First the data is added in the form of a text with headings. Malformed, small or ingredients of generic products were removed from the data. As the data was in heading-content format, it was changed to python dictionary of Ingredients, then all the multi-term ingredients were separated into different terms. Additional characters like ”[]”, ”()”, ”%”, ”,”, etc. as the semantic meaning of them would be unnecessarily complex. Next, we counted the occurrence of all the separated ingredients that are present and we got 1672 unique ingredients for both. Now the data is ready to be used for implementing various machine learning algorithm which has 1100 different food and beauty products.

C. One Hot Encoding

One hot encoding is a technique used to represent categorical data as numerical data, which can be used as input to machine learning algorithms. In one hot encoding, each category is represented by a binary vector of 0s and 1s, where the length of the vector is equal to the number of categories.

The cleaned data is now taken and intially before implementing PCA, top 112 separated ingredients were taken as features of data. If an ingredient was present in a product, the corresponding value for that product is changed to 1 representing the existence of the ingredient in the product. If it is 0, it represents absence of the ingredient in the product. This was directly fed into the model.

Later, the change was made to take all 1670 feature and then PCA is used for feature reduction and then the reduced vectors are passed to the model.

D. Algorithms

We have implemented following machine learning algorithms on our dataset for classification,

1) Decision Tree
2) Random Forest
3) Linear Regression
4) Logistic Regression
5) Naive Base
6) k-Nearest Neighbors
7) Support Vector Machine

The ROC curve and precision-recall curve are both commonly used evaluation metrics for binary classification models in machine learning. The ROC (Receiver Operating Characteristic) curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds.

We created ROC (Fig. 1) and Precision Recall Curves (Fig. 2) to evaluate and determine the most suitable algorithm for our dataset by comparing all the algorithms that we used.

From the above two graphs, we have decided that Support Vector Machine (SVM) is the most accurate model for our dataset.

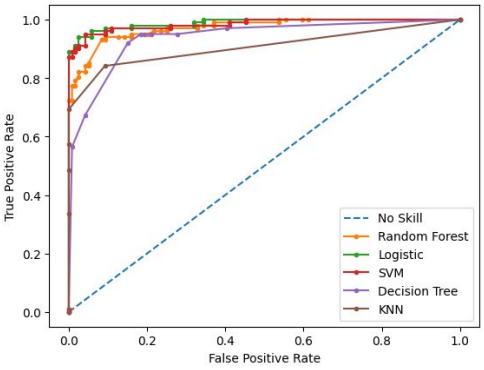

Fig. 1. ROC Curve

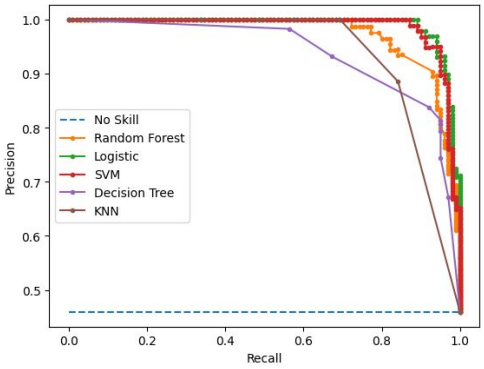

Fig. 2. Precision Recall Curve

E. One Model Selection

Support Vector Machine (SVM) is often considered better for binary classification due to its ability to find an optimal hyperplane that maximizes the margin between classes, leading to a clear decision boundary. SVM is also effective in handling non-linear data relationships through the use of kernel functions, allowing for more complex decision boundaries. Additionally, SVM is less prone to overfitting and performs well with limited training data, making it a robust choice for binary classification tasks.

In our binary classification model, SVM exhibited higher accuracy, precision, recall, and overall performance compared to the other algorithms, making it the optimal choice for our model. Moreover, SVM is more effective in high dimensional spaces. We have Implemented SVM algorithm using sk-learn library. Fig. 3 shows classification report for the same.

F. PCA

PCA is a technique used in machine learning and data analysis to reduce the dimensionality of data by identifying the most important information. It transforms high-dimensional data into a lower-dimensional representation, capturing the maximum variance. This technique is useful for visualizing and analyzing data, reducing noise, and improving the performance of machine learning models.

	precision	recall	f1-score	support
0	0.92	0.98	0.95	119
1	0.98	0.90	0.94	101
accuracy			0.95	220
macro avg	0.95	0.94	0.94	220
weighted avg	0.95	0.95	0.95	220

Fig. 3. SVM Before PCA

Run our dataset on PCA and got result as shown in figure 4. Got almost same accuracy with 1672 features in main dataset and with only 100 features after PCA.

	precision	recall	f1-score	support
0	0.90	0.99	0.94	158
1	0.99	0.90	0.94	172
accuracy			0.94	330
macro avg	0.94	0.94	0.94	330
weighted avg	0.95	0.94	0.94	330

Fig. 4. SVM After PCA

G. Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the best hyperparameter values for a machine learning model. GridSearchCV exhaustively tests a predefined set of hyperparameter values in combination, evaluates their performance using cross-validation, and returns the optimal hyperparameter values that yield the best model performance. We had implemented GridSearchCV on trained data after PCA. Got more accurate results then data after PCA shown in Figure 5.

	precision	recall	f1-score	support
0	0.95	0.97	0.96	175
1	0.97	0.94	0.95	155
accuracy			0.96	330
macro avg	0.96	0.96	0.96	330
weighted avg	0.96	0.96	0.96	330

Fig. 5. SVM After PCA

H. Cosine Similarity

Cosine similarity measures similarity between two vectors in multi-dimensional space. It calculates the cosine of the angle between the vectors, indicating their direction similarity. It ranges from -1 to 1, with 1 indicating perfect similarity, 0 indicating no similarity, and -1 indicating perfect dissimilarity. We had run Cosine Similarity from sk-learn library [5] and we got much desirable results. One result shown in Figure 6 where we got 10 most similar product of 'Patak's Aloo Tikki'.

```
input = "Patak's Aloo Tikki"

lst=recommend(input.lower())
lst
```

```
56                  patak's aloo tikki
54                  natan's aloo tikki
546       eastern chicken tikka masala
291                           mdh haldi
184                  haldiram's masala
55               bikanerwala aloo tikki
119    maggi hot and sweet chili sauce
464      mtr ready to eat chana masala
140                maggi chicken masala
463               mtr ready to eat chana
```

Fig. 6. Results of Cosine Similarity

RESULTS

I. Impact of number of components of PCA on Accuracy

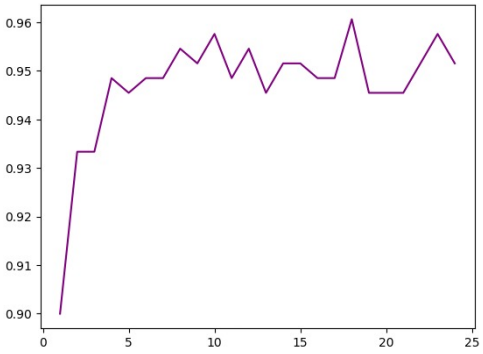

Fig. 7. Accuracy vs No. of Features

The graph of Accuracy vs. No. of components of PCA shows an irregular pattern. Taking first 3 to 4 components show good increase in accuracy of the model. After a certain number of components, the increase in accuracy is not clear.

Initially, the cumulative explained variance increases rapidly as the number of components increases, indicating that a small number of components capture a significant portion of the variance in the data. However, as more components are added, the rate of increase in cumulative explained variance slows down, resulting in a parabolic shape. This suggests that the additional components beyond a certain point contribute less to the overall explained variance in the data.

This type of graph is commonly observed in Principal Component Analysis (PCA), a technique used for dimensionality reduction in data analysis. It suggests that a smaller number of principal components can effectively capture the most important information in the data, while adding more

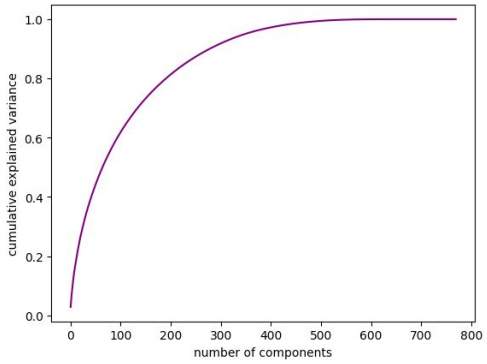

Fig. 8. Cumulative Explained Variance vs No. of Components

components may not necessarily lead to a significant increase in explained variance. Therefore, it may be prudent to select an appropriate number of components that strikes a balance between capturing enough variance and minimizing unnecessary complexity in the model.

J. Data Visulization of 2 feature data

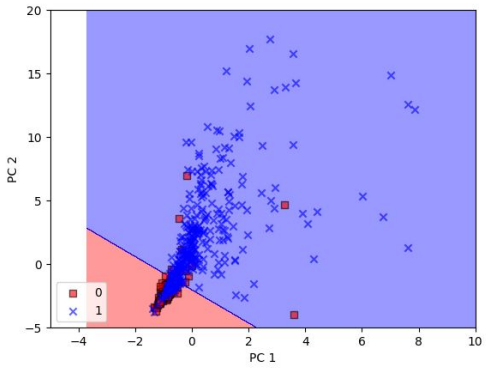

Fig. 9. Cumulative Explained Variance vs No. of Components

K. Checking Underfitting and Overfitting

We tested different train - test split ratios on the Support Vector Machine (SVM) model and observed the f1 score of the model. The test - train ratio ranged from 0.04 to 1.0. and 0.04 refers to almost all data used in training 1.0 refers to uninitialized weights. We observed that excluding the noise of random initializations each time of testing, a good ratio to prevent overfitting would be 20-80, and the model was less prone to overfitting.

CONCLUSION

In conclusion, our study thoroughly evaluated the performance of six different machine learning models, including Logistic Regression, Linear Regression, Naive Bayes, k-Nearest Neighbors, Decision Tree, and Random Forest. The evaluation was based on key metrics such as accuracy, F1-score, precision, and recall rate. Among these models, Support Vector Machine (SVM) stood out with optimal results, indicating its superior predictive ability and ability to minimize false positives.

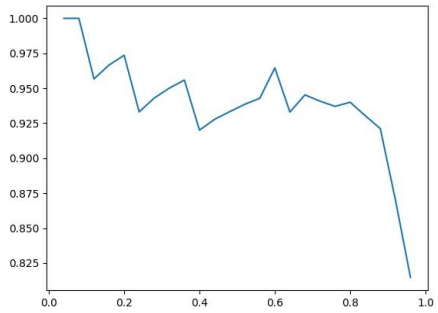

Fig. 10. Checking Underfitting and Overfitting

Further analysis revealed that using a train-test split ratio of 20:80 helped prevent overfitting in the SVM model, ensuring its robustness. We also employed several techniques to optimize the model, including reducing the dimensions of the dataset using Principal Component Analysis (PCA), tuning hyperparameters using GridSearchCV, and measuring the similarity between data points using cosine similarity.

Additionally, we implemented a recommendation model that provides a list of similar products based on the given input. This adds an extra layer of utility to our study, as it can aid in making personalized product recommendations to users.

Overall, our findings highlight the importance of selecting the right machine learning model and optimizing it for specific use cases to achieve accurate and reliable results.

SOURCE CODE

REFERANCE

[1] Swain, P. H., Hauska, H. (1977). The decision tree classifier: Design and potential. IEEE Transactions on Geoscience Electronics, 15(3), 142-147.

[2] Hosmer Jr, D. W., Lemeshow, S., Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley Sons.

[3] Dasu, T., Johnson, T. (2003). Exploratory data mining and data cleaning. John Wiley Sons.

[4] Pal, M. (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.