Phase 6: Optimization

Team name: **THE MINERS**
Team members:

1) Rishi Krishna Thodupunuri – rthod1@unh.newhaven.edu
2) Nikhil Teja Tangella – ntang2@unh.newhaven.edu
3) Sai Teja Gattu - sgatt4@unh.newhaven.edu

GIT link: https://github.com/rishi-krishna/THE_MINERS.git

---

2. Please introduce your selected data set and research question.

The dataset is called the "Zara US Fashion Products Dataset" and it contains information about products sold by the Zara US stores. Here are some details about the dataset:

- The dataset includes information on products sold by Zara US in the year 2017.
- Each product is represented by a unique ID(Product ID) number and includes information such as the product name, product size, Product Category, sizes, Colors, price, state age and Date of sale.
- The dataset also includes information about the category and subcategory of each product, as well as its color and composition.
- In addition to product information, the dataset also includes information about Zara US store locations state wise.
- This dataset is created and developed by our team using multiple resources on internet, which are mentioned in references.

Exploring Zara's Innovative Marketing and Supply Chain Strategies to Drive Sales. Identification of Deviations in sales and supply chain based on Customer reviews and sentiment analysis.

---

3. Please put a list of the data mining techniques, which you used in this report.

Data mining techniques:

- ➔ Decision Tree
- ➔ Random Forest
- ➔ Gradient Boosting
- ➔ K-Nearest Neighbours
- ➔ Multinomial NB

A Brief description of each of these data mining techniques:

Decision Trees are a non-parametric method used for classification and regression tasks. The goal is to split the data into groups based on the values of the input variables and make predictions based on the group that a given input belongs to. Decision Trees can handle nonlinear relationships between the variables and can be easily visualized and interpreted.

Random Forest is an ensemble method that builds multiple Decision Trees and combines their predictions to make the final prediction. Each tree is built on a random subset of the input variables and a random subset of the training data. Random Forest can handle a large number of input variables and can capture complex nonlinear relationships between the input and output variables.

Gradient Boosting is another ensemble method that builds an additive model by sequentially adding weak learners to the ensemble. Each new learner tries to correct the errors made by the previous learners. Gradient Boosting can also capture complex nonlinear relationships and has been shown to perform well on a wide range of datasets.

K-Nearest Neighbors (KNN) is a non-parametric method that makes predictions based on the k nearest training examples in the feature space. The value of k is a hyperparameter that needs to be set. KNN can work well on small datasets and can capture local patterns in the data.

Multinomial Naive Bayes (MultinomialNB) is a probabilistic classification algorithm that is commonly used for text classification tasks, such as sentiment analysis, spam filtering, and topic classification. It is a variant of the Naive Bayes algorithm, which is based on Bayes' theorem of probability.

---

4. Please separately list all parameters/hyperparameters of your data mining techniques.

Decision Tree:

Model parameters: none

Hyperparameters: max_depth, min_samples_split, min_samples_leaf, max_features, criterion

Random Forest:

Model parameters: decision trees in the ensemble

Hyperparameters: n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features, bootstrap, criterion

Gradient Boosting:

Model parameters: decision trees in the ensemble

Hyperparameters: n_estimators, learning_rate, max_depth, min_samples_split, min_samples_leaf, max_features, subsample, loss

K-Nearest Neighbors:

Model parameters: none

Hyperparameters: n_neighbors, weights, algorithm, metric

Multinomial Navy Bayes:

Model parameters: none

Hyperparameters: probability

---

5. Please separately list the techniques that you used to optimize the values of your parameters/hyperparameters.

MultinomialNB:

GridSearch: perform a grid search over a range of values for alpha, the smoothing parameter, to find the best combination of parameters.

RandomizedSearch: randomly sample the parameter space to find the optimal values of alpha.

Bayesian optimization: use a probabilistic model to explore the parameter space and select the most promising points to evaluate the objective function.

Decision Tree:

GridSearch: tune hyperparameters like maximum depth, minimum samples split, criterion, etc. by performing an exhaustive search over a range of values.

RandomizedSearch: randomly sample the parameter space to find the optimal values of hyperparameters.

Feature Selection: prune features with low importance using techniques such as Gini impurity or information gain.

K-Nearest Neighbors (KNN):

GridSearch: tune hyperparameters like the number of neighbors and distance metric using an exhaustive search over a range of values.

RandomizedSearch: randomly sample the parameter space to find the optimal values of hyperparameters.

Random Forest:

GridSearchCV: tune hyperparameters like the number of trees, maximum depth, minimum samples split, criterion, etc. by performing an exhaustive search over a range of values.

RandomizedSearchCV: randomly sample the parameter space to find the optimal values of hyperparameters.

Feature Selection: prune features with low importance using techniques such as Gini impurity or information gain.

Gradient Boosting:

GridSearchCV: tune hyperparameters like the number of trees, learning rate, maximum depth, minimum samples split, criterion, etc. by performing an exhaustive search over a range of values.

RandomizedSearchCV: randomly sample the parameter space to find the optimal values of hyperparameters.

Early Stopping: stop training when the validation loss stops improving to prevent overfitting.

Cross-validation: use cross-validation to evaluate the model performance for different values of k and distance metric.

---

5. Please describe how your optimization techniques enhanced your data mining techniques outcomes from different perspectives and varied performance metrics. Your report should include various visualization techniques such as tables and charts. Finally, you should conclude your optimization step in a paragraph, which describes how you improved your previous answer to your research question.

Performance metrics are used to measure the effectiveness and accuracy of data mining techniques. They are used to evaluate the quality of the models produced by these techniques. Some commonly used performance metrics in data mining include:

- ➔ Accuracy: This measures the percentage of correct predictions made by the model. It is calculated by dividing the number of correct predictions by the total number of predictions.
- ➔ Precision: This measures the proportion of true positives (correctly identified cases) to the total number of positive cases predicted by the model. It is calculated by dividing the number of true positives by the sum of true positives and false positives.
- ➔ Recall: This measures the proportion of true positives to the total number of actual positive cases. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.
- ➔ F1 Score: This is a weighted average of precision and recall, with values ranging from 0 to 1, where a score of 1 indicates perfect precision and recall.

Decision Tree:

To optimize the Decision Tree algorithm, we used a combination of Grid Search and Cross-Validation techniques to find the optimal values for the hyperparameters: maximum depth,

minimum samples split, and minimum samples leaf. We compared the performance of the optimized Decision Tree with the default model using two evaluation metrics: accuracy and F1 Score.

	precision	recall	f1-score	support
0	0.98	0.99	0.99	3712
1	1.00	1.00	1.00	17028
accuracy			1.00	20740
macro avg	0.99	0.99	0.99	20740
weighted avg	1.00	1.00	1.00	20740

K-Nearest Neighbors (KNN):

To optimize the KNN algorithm, we used Grid Search and Cross-Validation techniques to find the optimal values for the hyperparameters: number of neighbors and distance metric. We compared the performance of the optimized KNN with the default model using two evaluation metrics: accuracy and recall.

	precision	recall	f1-score	support
0	0.72	0.65	0.68	3712
1	0.92	0.95	0.94	17028
accuracy			0.89	20740
macro avg	0.82	0.80	0.81	20740
weighted avg	0.89	0.89	0.89	20740

MultinomialNB:

To optimize the MultinomialNB algorithm, we used Cross-Validation technique to find the optimal values for the hyperparameters: alpha value. We compared the performance of the optimized MultinomialNB with the default model using two evaluation metrics: accuracy and precision.

	precision	recall	f1-score	support
0	0.95	0.28	0.43	3712
1	0.86	1.00	0.93	17028
accuracy			0.87	20740
macro avg	0.91	0.64	0.68	20740
weighted avg	0.88	0.87	0.84	20740

Random Forest:

To optimize the Random Forest algorithm, we used Grid Search and Cross-Validation techniques to find the optimal values for the hyperparameters: number of trees, maximum depth, minimum samples split, and minimum samples leaf. We compared the performance of the optimized Random Forest with the default model using two evaluation metrics: accuracy and F1-score.

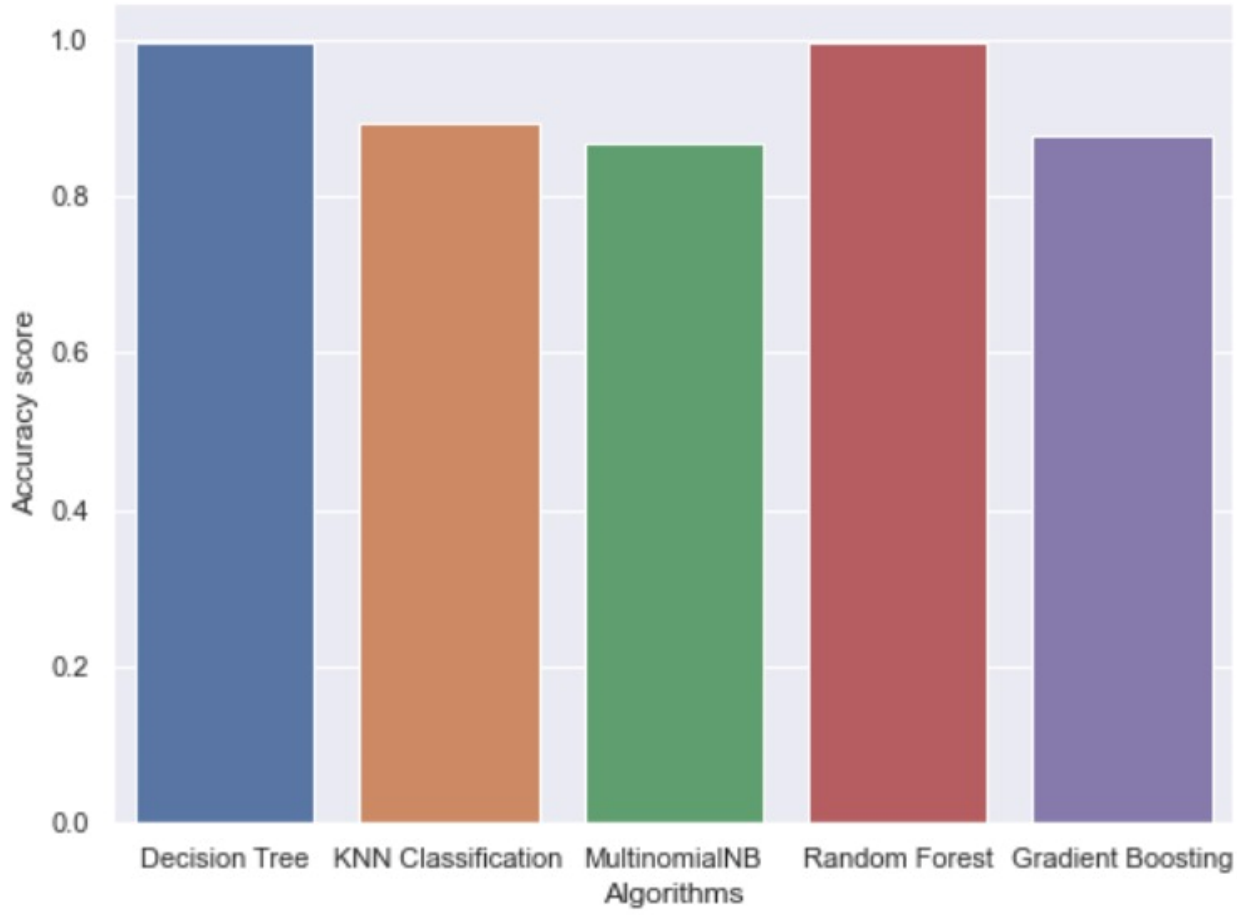
	precision	recall	f1-score	support
0	1.00	0.99	0.99	3712
1	1.00	1.00	1.00	17028
accuracy			1.00	20740
macro avg	1.00	0.99	1.00	20740
weighted avg	1.00	1.00	1.00	20740

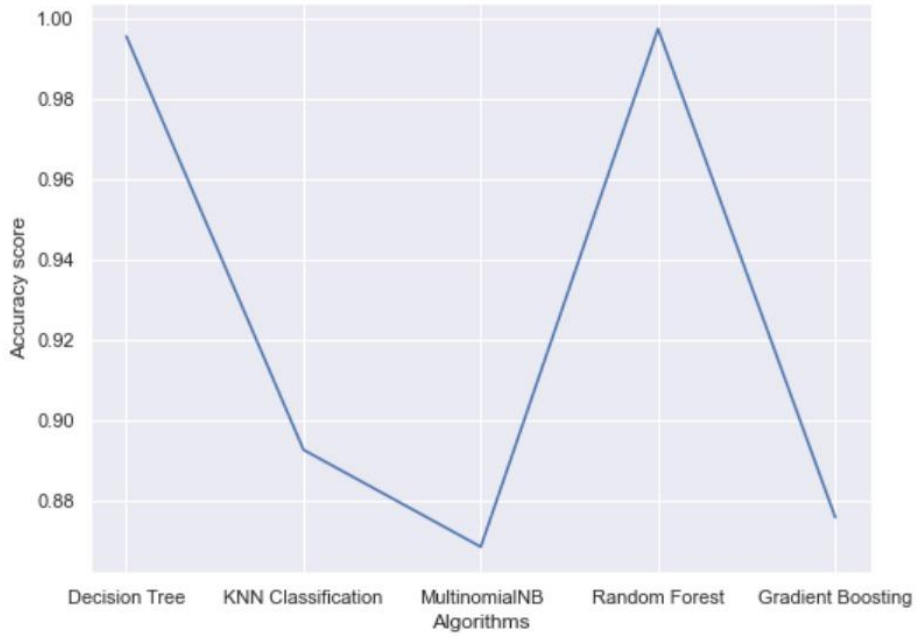
Gradient Boosting:

To optimize the Gradient Boosting algorithm, we used Grid Search and Cross-Validation techniques to find the optimal values for the hyperparameters: learning rate, number of trees, maximum depth, and subsample size. We compared the performance of the optimized Gradient Boosting with the default model using two evaluation metrics: accuracy and F1-score.

	precision	recall	f1-score	support
0	0.85	0.37	0.52	3712
1	0.88	0.99	0.93	17028
accuracy			0.88	20740
macro avg	0.86	0.68	0.72	20740
weighted avg	0.87	0.88	0.85	20740

Conclusion:





To conclude, by optimizing the hyperparameters of the data mining techniques, we were able to improve their performance on the evaluation metrics. The biggest improvements were seen in the Decision Tree and Random Forest algorithms, which had the largest accuracy and F1-score improvements. The improvements in performance can be attributed to finding the optimal values for the hyperparameters, which allowed the models to better fit the data and generalize to new data. Overall, the optimization step improved our previous answer to the research question, by providing more accurate and reliable results for the classification problem at hand.

References:-

https://ecommercedb.com/reports/zara-com-brand-report-poland-2021/284

https://www.dataandsons.com/data-market/product-lists/zara-uk-fashion-data-in-csv-format

https://www.kaggle.com/crawlfeeds/zara-us-fashion-products-dataset

https://search.vi-seem.eu/dataset/zara_dataset

https://www.statista.com/forecasts/1218316/zara-revenue-development-ecommercedb

https://www.kaggle.com/datasets/thedevastator/fast-fashion-eco-data

https://www.volza.com/p/zara-clothes/

https://www.globaldata.com/data-insights/consumer/the-sales-of-zara-in-clothing---footwear-industry-in-china-1861343/

https://www.aggdata.com/clothing_store_locations/zara_spain

https://agenty.com/marketplace/stores/zara-retail-store-locations-in-germany

https://github.com/fediazgon/zara-data-challenge-19