PROJECT MILESTONE 2

DATA MODELING

DATA SCIENCE II

COSC 4337

Members

Luong Thuy Ngo (2048659)

Syaneth Khatt (2114829)

Karla Castello (2138671)

QUICK RECAP OF THE DATASET

Dataset: Customer Personality Analysis

- ➢ The selected dataset is the Customer Personality Analysis dataset, sourced from the Kaggle repository. With this data, we want to see common patterns of customers to see what type of customers is more likely to buy more from this company and determine certain strategies to target these customers in future campaign

Dataset link:

https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis?resource=download

MODELS

A classification model attempts to draw some conclusions from observed values. Given one or more inputs, a classification model will try to predict the value of one or more outcomes. Outcomes are labels that can be applied to a dataset. In our previous report, we used unsupervised techniques to clean our data, but since there was a class label in our original dataset, we are using supervised models to learn more about our data.

1. Decision Tree Classifier

The Decision Tree Classifier selects the best attributes to split the data and makes that attribute a decision node to break the dataset into smaller subsets. The Decision tree is then built recursively by repeating the process for each child node. The attribute selection measure is used to select the splitting criterion to split the data in the best possible way. The different measures include Entropy, Information Gain, Gini index, Gain Ratio, and Reduction in Variance. Using this model, it will be easy to classify new data by making decisions going down the tree.

2. Random Forest Classification

The second model used is very similar to the decision tree classifier, in which it creates many decision trees and labels them with a class, and the class with the most labels becomes the model's prediction. The basic difference being it does not rely on a singular decision. It assembles randomized decisions based on several decisions and makes the final decision based on the majority.

3. **Logistic Regression**

Logistic regression is similar to linear regression but is used to model the probability of a finite number of outcomes, typically two. Logistic regression predicts the output of a categorical dependent variable. In essence, a logistic equation is created in such a way that the output values can only be between 0 and 1. These output values are the probability of an event occurring, based on a given dataset.

HYPERPARAMETER TUNING

In order to get the best possible model for our dataset, hyperparameter tuning is needed for each model. Hyperparameters are used to structure the model and get the wanted model parameters. Since the values for the hyperparameters cannot be estimated by the data, we manually insert possible values for all hyperparameters. To automize the process of finding the best values for each hyperparameter, we used GridSearch cross-validation.

Hyperparameters we used:

1. **Decision Tree Classifier**

- max_depth = [10, 20, 30, 40]
- criterion = ['gini', 'entropy']

2. **Random Forest Classification**

- n-estimators = [10, 50, 100]
- criterion = ['gini', 'entropy']
- max_features = ['auto', 'sqrt', 'log2']
- max_depth = [10, 20, 30, 40]

3. **Logistic Regression**

- solvers = ['newton-cg', 'lbfgs', 'liblinear']
- penalties = ['l1', 'l2', 'elasticnet']
- c_values = [100, 10, 1.0, 0.1, 0.01]

The best hyperparameters in each model:

NO CROSS-VALIDATION

1. Decision Tree Classifier

Best hyperparameters found on the development set for Decision Tree:

{'criterion': 'gini', 'max_depth': 10}

2. Random Forest Classifier

Best hyperparameters found on the development set for Random Forest Classifier:
{'criterion': 'entropy', 'max_depth': 10, 'max_features': 'log2', 'n_estimators': 100}

3. Logistic Regression

Best hyperparameters found on the development set for Logistic Regression:
{'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}

Using all of the hyperparameters we chose and running them all in the GridSearch Function, we found the best hyperparameters for each model without accounting for overfitting. For Decision Tree Classifier, the best splitting criterion is the Gini index, and the best max depth of the tree is 10 nodes. Although the nodes will not be pure, it still can give us an idea of their classification. For Random Forest Classifier, the best splitting criterion is entropy, which is the impurity score of each node. The maximum depth of the tree is also 10 nodes. The maximum number of features to consider when looking for the best split is log2 of the total number of features. The best value for the ‘n_estimators’ parameter is 100, which is the number of trees in the forest. Lastly, for Logistic Regression, the C parameter is 0.1, meaning the model will have strong regularization. The penalty parameter becomes L2, which is a type of regularization that adds the squared magnitude of the coefficients as the penalty value to the loss function. For the last parameter in logistic regression, liblinear was chosen for the solver parameter, which is the algorithm to use in the optimization problem.

WITH STRATIFIED CROSS-VALIDATION

1. Decision Tree Classifier

Best hyperparameters found on the development set for Decision Tree:
{'criterion': 'entropy', 'max_depth': 10}

2. Random Forest Classifier

Best hyperparameters found on the development set for Random Forest Classifier:
{'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'n_estimators': 100}

3. Logistic Regression

Best hyperparameters found on the development set for Logistic Regression:
{'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}

In order to prevent overfitting on the models, we used Stratified Cross-Validation. In Stratified Cross-Validation, we use stratified sampling to make sure that the proportion of the feature of interest is the same in the original, training and testing set. This is to make sure that no value is over or under-represented in the training and testing sets. For the decision tree classifier, the hyperparameter ‘criterion’ changed after cross-validation from ‘gini’ to ‘entropy.’ This means that the entropy measure was the best splitting criterion for the decision tree. The maximum depth of the tree stayed the same. For the Random Forest Classifier, all of the hyperparameters stayed the same except for the ‘max_features’ parameter. In this case, the ‘max_features’ are set to ‘auto’, which means that instead of a specific number of features in a single tree, it will take all of the features that make sense in every tree. The best hyperparameters for Logistic Regression stayed the same after cross-validation.

PERFORMANCE EVALUATION

THE DATASET WITHOUT CROSS-VALIDATION

	Decision Tree Classifier	**Random Forest Classifier**	**Logistic Regression**
Confusion Matrix	[[349 26] [46 21]]	[[372 3] [49 18]]	[[368 7] [45 22]]
F-1 SCORE	0.6374572795625427	0.6718821379625399	0.6961717428087987
Area Under the ROC Curve (AUC-ROC)	0.6220497512437811	0.6303283582089552	0.6548457711442787
Root Mean Squared Error (RMSE)	0.4036036763977875	0.3429971702850177	0.3429971702850177
R2_score	-0.2666268656716422	0.08521393034825842	0.08521393034825842

COMMENT AND EXPLANATION

By looking at the outputs above of each model, we obtain the highest F1-score for Logistic Regression among all three models. This shows precision and recall scores are maximized for the most accurate data in Logistics Regression with F1-score of 0.696 followed by Random

Forest Classifier 0.672 and Decision Tree Classifier 0.637. Additionally, Logistic Regression has lower false positive and false negative while Decision Tree has a higher false positive and false negative in their respective confusion matrices. With this being said, both precision and recall scores for Logistic Regression are higher than the Decision Tree classifier (see the calculation below).

Logistic regression:

- Precision: 368/(368+7) = 0.98
- Recall: 368/(368+46) = 0.89

Decision Tree:

- Precision: 349/(349+26) = 0.93
- Recall: 349/(349+46) = 0.88

We also observe that the results of the area under the ROC curve are pretty similar across all models. y are a little over 0.6 which is in the range of acceptable discrimination.

It can be noticed that the Root Mean Squared Error(RMSE) for Logistic Regression and Random Forest are the same and about 0.06 lower than the RMSE of the Decision Tree classifier. By computing RMSE, we hope to see a result that is much closer to 0 or as lower as possible so that it can provide the best accuracy. From the table, the RMSE of Decision Tree is about 40% which provides a higher average model prediction error than the other two models and potentially concludes that its performance is not best predicting the data. Similarly, we received the same R2 score = 0.0852 for both Random Forest Classifier and Logistic Regression, showing that each model has positive correlations between y_test and y_predict. While Decision Tree classifier has an R2 score = -0.266 below 0 which illustrates that the model is predicting worse than the mean of the target values and it has a negative correlation between y_test and y_predict.

Without using cross-validation, we attain superior outputs in all categories for Logistic Regression than Random Forest Classifier and Decision Tree Classifier. However, the AUC and R2-score of Random Forest and Logistic Regression are the same and slightly different in F1 Score. This reflects a similar data performance between these two models and more accurate execution. We recognized worse performance from Decision Tree Classifier especially higher results in Root Mean Squared Error and negative results of R2 score.

THE DATASET WITH CROSS-VALIDATION

	Decision Tree Classifier	Random Forest Classifier	Logistic Regression
Confusion Matrix	[[1849 28] [67 266]]	[[1877 0] [83 250]]	[[1823 54] [175 158]]
F-1 SCORE	0.9117193554314567	0.9180007250876296	0.7603598697839598
Area Under the ROC Curve (AUC-ROC)	0.8919406886908219	0.8753753753753754	0.7228525808706949
Root Mean Squared Error (RMSE)	0.2073316795363567	0.19379515237996242	0.32190046520976395
R2_score	0.6641020349065101	0.7065312515498983	0.19030911572200848

COMMENT AND EXPLANATION

Confusion Matrices across all three models using stratified cross-validation are noticeably higher than the first table (without cross-validation). We obtain similar results of F1-Score approximately about 0.9 between Decision Tree Classifier and Random Forest Classifier while Logistic Regression is about 0.15 lower. This shows an incredible performance for best accuracy from Random Forest and Decision Tree. This pattern also applies to Area Under the ROC Curve (AUC-ROC), where the best score achieved is approximately 0.89 for Decision Tree Classifier followed by Random Forest and Logistic Regression. However, Random Forest Classifier has the highest R2_score, and the Decision Tree model comes just a tiny bit behind whereas we can see the drastic opposite of Logistic Regression, but thankfully, it is not below 0 which indicates there is a positive relationship between y_test and y_predict. Likewise, since both Decision Tree and Random Forest performed quite well resulting in the confusion matrix, F1 score, AUC, and R2_score, we also attained Root Mean Squared Error the lowest for these two models within this dataset which means it aligns with all results in all listed categories above. We assume the Decision Tree classifier executed the best accuracy among all models using Cross-Validation.

CONCLUSION OF BOTH WITH AND WITHOUT CROSS-VALIDATION

By comparing the two tables above, it can be seen that the performance of data with stratified cross-validation should be put to better use in achieving more accuracy. We used metrics such as confusion matrix, F1_Score, AUC_ROC, Root Mean Squared Error, and R2_Score to find which model is the best fit and give the accurate output. From this observation, we see a significant contrast between Decision Tree Classifier and Logistic Regression. In the first table (without cross-validation), Decision Tree Classifier performed very badly, and RSME was pretty high whereas Random Forest and Logistic Regression were the opposite and are assumed to have a very close margin in performance. As a result, we concluded that Logistic Regression is the most effective method for accomplishing the best accuracy without utilizing Cross-Validation. In contrast, when we used stratified cross-validation to prevent overfitting, we obtained the most accurate within the Decision Tree Classifier followed by Random Forest which was just a little bit behind Decision Tree in all metrics that we used. However, Logistic Regression surprisingly performed the worst among all three methods**.** We saw that the Decision Tree classifier and Logistic Regression fluctuated and contrasted their own values when cross-validation was added. After a thorough observation, we decided that Random Forest Classifier is the best-fit model for accuracy. Even though it came second in rank with and without using cross-validation but its performance and output values are always close to the model that ranked first and more stable for accurate execution.

REFERENCES

DataCamp. (n.d.). "Decision Tree Classification in Python". https://www.datacamp.com/tutorial/decision-tree-classification-python.

Fernandes, A. (2020). Towards Data Science. "What is Stratified Cross-Validation in Machine Learning?" https://towardsdatascience.com/what-is-stratified-cross-validation-in-machine-learning-8844f3e7ae8e#:~:text=Cross%2Dvalidation%20implemented%20using%20stratified,set%20and%20the%20test%20set.

IBM. (n.d.). "What is logistic regression?" https://www.ibm.com/topics/logistic-regression#:~:text=Resources-,What%20is%20logistic%20regression%3F,given%20dataset%20of%20independent%20variables.

Patel, Akash. , Kaggle, 2022, "Customer Personality Analysis"

www.kaggle.com/datasets/imakash3011/customer-personality-analysis.

PennState: Statistics Online Courses, "11.6 - Example: Places Rated after

Standardization: STAT 505"

https://online.stat.psu.edu/stat505/lesson/11/11.6

Sarkar, A. (2017). Understanding Random Forest. Towards Data Science. Retrieved from

https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

Shengping Yang, Gilbert Berdine MD. "The receiver operating characteristic (ROC) curve" https://pulmonarychronicles.com/index.php/pulmonarychronicles/article/view/391/848#:~:text=In%20general%2C%20the%20rule%20of,Poor%20discrimination

INSTRUCTIONS TO RUN THE CODE

1. Download the dataset from this link:

https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis/download?datasetVersionNumber=1

2. Extract the zip folder to get the file named ‘marketing_campaign.csv’
3. Place the dataset into the same folder as the file named ‘notebook_firstDeliverable.ipynb’
4. Open the file ‘notebook_firstDeliverable.ipynb’ with Jupyter Notebook
5. Run the code (will need to download the necessary libraries if it is not pre-installed on your computer)