Logan Heft
Data 1030 Midterm Report (12/4/2021)
https://github.com/lheft/Midterm

INTRODUCTION

This dataset is the biographic information of potential customers at a Portuguese bank from a direct marketing campaign, such as age, job, marital status, loan balance, etc. The feature variable that is in this dataset is labeled as 'y'. The bank that created this dataset was attempting to sign up customers for this new product which was an attractive long term deposit that offered "very good" rates. The 'y' variable indicates whether or not they were able to get the potential customer to sign up or not. It is a binary data field with the labels of 'yes' and 'no'. This problem is very clearly a classification problem, due to the fact that the target variable is binary and not a continuous value. As for the potential models that may apply to the dataset: Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, or K-Nearest Neighbors would all apply. Any one of these models, if built correctly, could conceivably save the bank massive amounts of time and money that would usually be spent on customers that they had no chance of signing up for a new product.

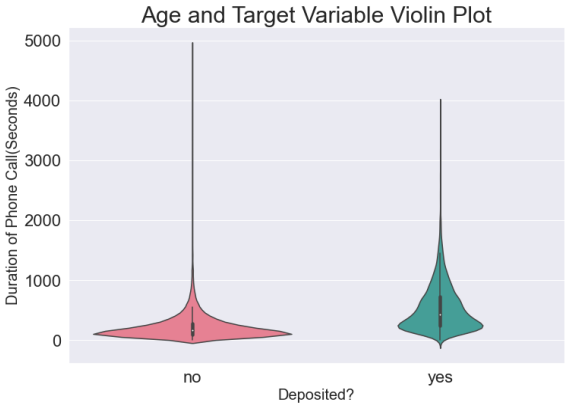
This data is quite interesting because it has many different data types and it is also thee original data from an actual campaign. For example, it has a nominal job variable, a few binary variables, as well as many continuous loan balance data. Having so many data types will require a lot of preprocessing work. There are approximately 45,000 data points and 17 features if we include the target variable: **Age**: This is how old the individual is in years(Numeric >=18), **Job**: Current occupation *['admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown']*, **Marital**: This is their marital status which includes married, single, or divorced (with divorced also including those who are widows), **Education**: Highest level of education achieved *['primary', 'secondary', 'tertiary', 'unknown']*, **Default**: Does the individual currently have defaulted credit (Binary: yes/no), **Balance**: average yearly balance in euros (numeric), **Housing**: Does the individual have a housing loan (Binary: yes/no), **Loan**: Does the individual have a personal loan, **Contact**: How was the individual contacted? *['cellular', 'telephone', 'unknown']*, **Day**: Number of days since this individual has last been contacted during this campaign, **Month**: Month that the individual was last contacted *['jan', 'feb','mar',etc.]*, **Duration**: Duration (denoted in seconds) of the last point of contact, **Campaign**: number of contacts performed during the campaign for this individual, **pdays**: number of days that passed since client was last contacted (-1 means that the client was not previously contacted), **Previous**: number of contacts performed before this campaign,**Outcome**: outcome of previous marketing campaign *("unknown", "other", "failure"," success")*, **Y**-has the client subscribed to the term deposit(Binary target variable: yes/no).

This data set was originally published by Sergio Moro, Raul Laureano, and Paulo Cortez, all of which being staff or students at the Instituto Universitario de Lisboa (Lisbon University Institute). Linked here is their original paper regarding the first use of their dataset, which also includes information about how the data was collected and processed before being published. Here they explain how they had originally had a larger dataset than what they published (55817 rows instead of 45211), however many of the rows were missing data and they therefore opted to discard such instances citing that many ML models do not work

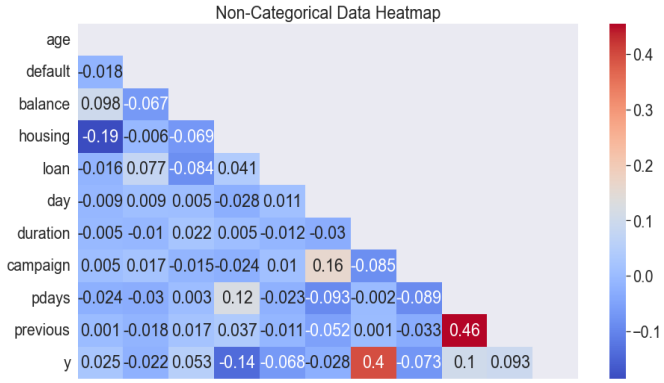
well with missing data. Another relevant piece of information from their article is that there used to be 59 client attributes (features), but they again opted to delete some of these features before publishing the dataset, citing that too much of the information was missing and/or redundant. The researchers were ultimately very successful with their work. They published 3 iterations of a Naive Bayes, Decision Tree, and Support Vector Machine Model. In that same order they achieved these AUC (Area Under the ROC Curve scores) respectively: .870, .868, and .938. As illustrated with a AUC score of .938 they were extremely successful in their algorithm creation, and provide a benchmark that I can hope to surpass. There have also been multiple individuals on Kaggle that have created Machine Learning models and posted their work, but none have been as successful as the original creators of the dataset with success being defined by their AUC scores.

EDA

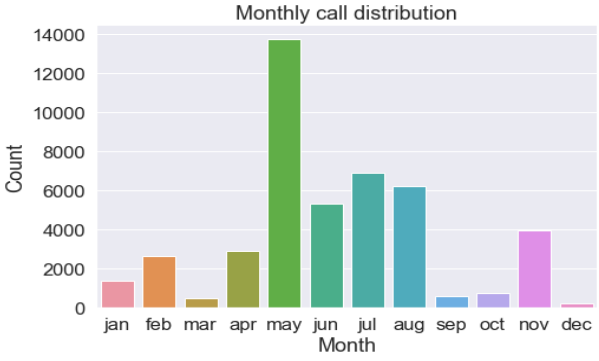
With so many different features and feature data types, a lot of EDA was required to begin to understand the data. For example, in the violin plot to the right, it appears that distribution of yes and no for the target variables is rather symmetric over ranges of age, however in the yes distribution there seems to be a slight increase when the individual is over 60 years of age. This could suggest that age could play a large role in our future ML models.



In figure 2, there is a heat map that displays the correlation of some of the variables (It is important to note that housing, loan, campaign and previous are all binary variables, with 1 representing yes and 0 representing no). We see that none of the variables have a very strong correlation, with the highest absolute value of any correlation being only .46 between number of days since the client was last contacted and number of times the customer was contacted prior to this call. This is extremely important to know because it ensures us that our model will not be affected by multicollinearity.



Lastly, in figure 3 we have the distribution of how many calls occurred each month. Interestingly there is a massive spike in the number of calls in the month of May, more than twice as many as the next highest calls in a month. Notably there are also 4 months with extremely low amounts of calls.. As the data is further explored we hope to discern the reasonings behind these disparities in calls.



Methods

After the EDA, the preprocessing of the data began. This dataset is considered balanced and this is because there are 45211 total data points, with 5289 being marked as yes for the target variable and 39922 being marked as no, which does not meet the threshold of being considered imbalanced. Approximately 13% of the data is a yes, which is above the 5% or less threshold to mark data as imbalanced. The data is also considered IID therefore we perform a simple k-fold split. The number of splits was set to 4 (X_other, X_test, Y_other, Y_test). The way the data was preprocessed was by first replacing all the yes and no's with 1's and 0's. I then applied a OneHotEncoder for all the features with the object data types which included fields like marital status, month, and jobs. A MinMax encoder was applied to all numeric values such as: age, day, campaign, pdays, and previous fields. After preprocessing the dataset has 51 features. We included all of the preprocessing within a GridSearch function which also included the hyperparameter grid we wanted to hypertune and as well as our scoring which was set to roc_auc. The grid search would run 5 times, each time using a different random state. When the grid search was done running, the best hyperparameters from each random state would be printed out along with the roc_auc scores. Jason Brownlee's article "Hyperparameter Optimization with Random Search and Grid Search" served as the template for which hyperparameters and in what range we'd be searching in.

As seen above, for the 3 grid searches for logistic regression the hyperparameters {C=100, max_iter=500,

```
Fitting 4 folds for each of 25 candidates, totalling 100 fits
best model parameters: {'logisticregression__C': 100, 'logisticregression__max_iter':
500, 'logisticregression__solver': 'liblinear'}
roc auc score: 0.9049602092164639
Fitting 4 folds for each of 25 candidates, totalling 100 fits
best model parameters: {'logisticregression__C': 100, 'logisticregression__max_iter':
500, 'logisticregression__solver': 'liblinear'}
roc auc score: 0.9072603022713495
Fitting 4 folds for each of 25 candidates, totalling 100 fits
best model parameters: {'logisticregression__C': 100, 'logisticregression__max_iter':
500, 'logisticregression__solver': 'lbfgs'}
roc auc score: 0.9055653367028254
```

solver='liblinear'} were returned twice, while {solver='lbfgs'} was returned only once, therefore our final logistic regression model is going to be trained using the first set of parameters. The final model was trained this time using the newly found hyperparameters and another new random state. Each parameter also prints out its auc roc score. It was decided to use AUC ROC vs Accuracy as the final evaluation metric because while our data doesn't technically meet the threshold of approximately 5% to be considered imbalanced, it comes quite close. Meaning that our models would be able to get very high accuracy scores by predicting that all observations belong to the majority class.

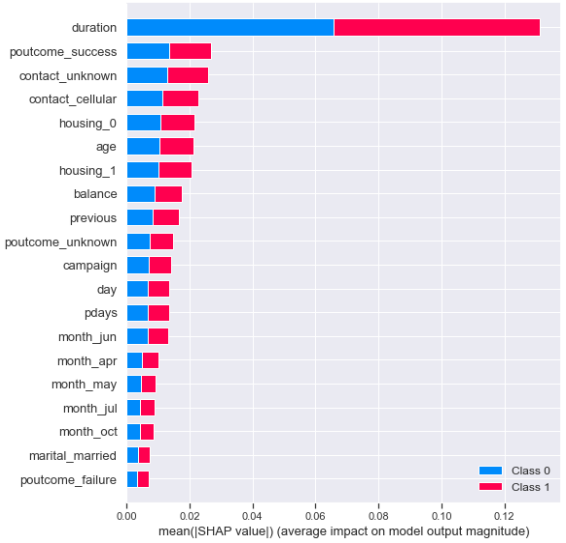
Results

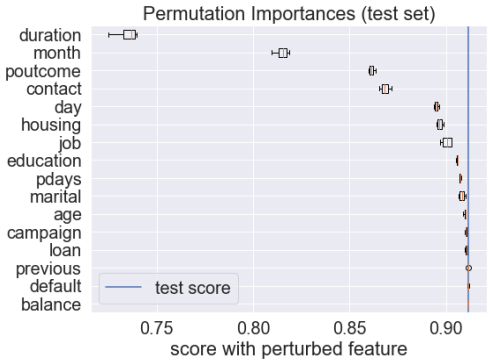
After the model was finished training, a classification report was printed out which displayed information such as: precision, recall, f-1 score, accuracy, macro avg, and weighted average. In the report there were 6 models tested: Logistic Regression, Decision Tree, Random Forest, KNearestNeighbor Classifier, GradientBoostingClassifier, and SVC. The results can be seen to

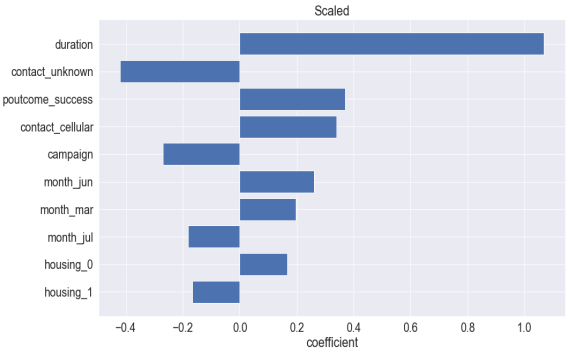
Model	ROC AUC Score
Random Forest	0.930262
XGB	0.924567
SVC	0.902835
Logistic Regression	[0.8090391689679755]
KNeighbors	0.80699
Decision Tree	[0.6936481178163977]

the right in a table displaying the final models AUC ROC Scores. It's difficult to measure whether or not these ROC AUC scores are “good” or not. While it's true that a higher score is better than a lower score, it is very application dependent. According to Matt Krause on Github, an AUC score of .95 while trying to identify handwritten digits is far below the industry standard, while an AUC score of .8 while trying to identify profitable investments would make you a very rich individual. The scores to the right are mostly useful for comparing the models to each other and to models other individuals have created on Kaggle. With this being said, the Random Forest Model and XGBClassifier model beat out many of the models created on Kaggle where this dataset was first found.

Looking at the feature importance (represented by the SHAP value) of the models we see duration being the most important feature in most of the models, and by a margin of 6x the next closest feature. The duration variable represents how long the phone call was between the marketer and the potential customer. While there was a correlation between higher durations and a customers willingness to try the new product, this does not mean causation, however workers at the bank may want to emphasize trying to keep customers on the phone as long as possible. Another feature(s) that had high SHAP values in many of the models was the various month dummy variable features. When originally looking at the data, it was easy to dismiss month features as it did not seem likely that the time of year when called would have any significance. It is possible however that holidays, tax season, pay days, etc. that happen during these months are more influential then the actual months themselves. Surprisingly, in the perturbed feature tests, balance and default had the least impact for most models, while less surprising is that duration remained the most important (Pictured on the right is the Perturbed Feature scores for the SVC model). The last figure on this page shows the coefficients with the highest absolute values of the final logistic regression model, which is another measure of feature importance. The coefficients of a logistic regression can be used as a measure of feature importance only if all features have a zero mean and the same standard deviation therefore to create this image the features had to be specially preprocessed using a standard scaler.







Outlook

The weak spots of this analysis is that it was extremely limited by the computational resources available. For example, while grid searching, there were only 3-5 loops searches for each model type. Originally there were 11, however, the time to run the grid search was becoming extensive, hindering productivity. Another computationally intensive task was finding the SHAP values. Had there been access to faster computers or even if there were

less of a time restraint, more hyperparameter ranges could have been explored, and more models tested. Access to faster computers would have also allowed for the exploration of feature generation using algorithms that utilize Friedman's H statistics which looks at the partial dependence function of features, but this is extremely computationally intensive. Iit would also be interesting to do some statistical tests to establish whether the difference between model skill is due to being a better model or if it was a product of chance and randomness.

References

-armatitaarmatita. “Fine Control over the Font Size in Seaborn Plots for Academic Papers.” *Stack Overflow*, 5 Mar. 2016, https://stackoverflow.com/questions/36220829/fine-control-over-the-font-size-in-seaborn-plots-for-academic-papers/36222162.

-Brownlee, Jason. “Tune Hyperparameters for Classification Machine Learning Algorithms.” *Machine Learning Mastery*, 27 Aug. 2020, https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/.

Molnar, Christoph. “Interpretable Machine Learning.” *8.3 Feature Interaction*, 11 Nov. 2021, https://christophm.github.io/interpretable-ml-book/interaction.html.

-Moro, Sergio, and Paulo Cortez. *Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology*. https://repositorium.sdum.uminho.pt/bitstream/1822/14838/1/MoroCortezLaureano_DMApproach4DirectMKT.pdf.

Okamura, Scott. “GRIDSEARCHCV for Beginners.” *Medium*, Towards Data Science, 30 Dec. 2020, https://towardsdatascience.com/gridsearchcv-for-beginners-db48a90114ee.

-Patra, Sushovan. “Bank Marketing Campaign.” *Kaggle*, 14 Sept. 2021, https://www.kaggle.com/edith2021/bank-marketing-campaign/code.

-Sreenivasan, Hard. *Is It Better Using Training-Test Split or K-Fold CV, When ...* 11 June 2017, https://www.quora.com/Is-it-better-using-training-test-split-or-k-fold-CV-when-we-are-working-with-large-datasets.

-“Statistical Data Visualization.” *Seaborn*, https://seaborn.pydata.org/.

-Tavory, Ami. “In Supervised Learning, Why Is It Bad to Have Correlated Features?” *Data Science Stack Exchange*, 7 Nov. 2017, https://datascience.stackexchange.com/questions/24452/in-supervised-learning-why-is-it-bad-to-have-correlated-features.

-“Visualization with Python.” *Matplotlib*, https://matplotlib.org/