Udacity Data Science Capstone Technical Blog Post

Creating a Customer Segmentation Report for Arvato Financial Services

High Level Overview

In today's hyper-competitive business landscape, understanding your customer base is paramount. In our latest project, we embark on a journey to delve deep into demographics data for customers of a mail-order sales company in Germany. Our mission? To identify the individuals most likely to become loyal customers. To achieve this goal, we draw comparisons between this customer data and the broader population of Germany. This exciting project unfolds across several stages, each designed to illuminate the path to effective customer acquisition and targeted marketing through Data Exploration and Preparation, Unsupervised Learning and Supervised learning Algorithms

Understanding the Data and Strategy for Finding a Solution

Description of Input Data

The initial input data consists of two files, Udacity_AZDIAS_052018.csv, which is the general population data along with all of the relevant attributes and contains 891,221 records with 366 columns. Udacity_CUSTOMERS_052018.csv is the subset of the population that is customers and contains 191,652 rows with 369 columns. Each column varies in its datatype and can be anything from an integer to a float to an object. The goal of these columns is to help us identify what attributes are most predictive in determining if someone is likely to become a customer of the business. These attributes describe the individuals that make up the population and a few examples of what is available to us are:

- A person's academic background
- Information about their residence
- Their affinity for things like culture, religion and overall mindedness
- Purchasing habits for different categories of goods and services

Once all of this data is understood, we can use our findings to train different machine learning algorithms and apply this knowledge to test data to predict if a person is more or less likely to be converted to a customer.

Strategy to Solve the Problem

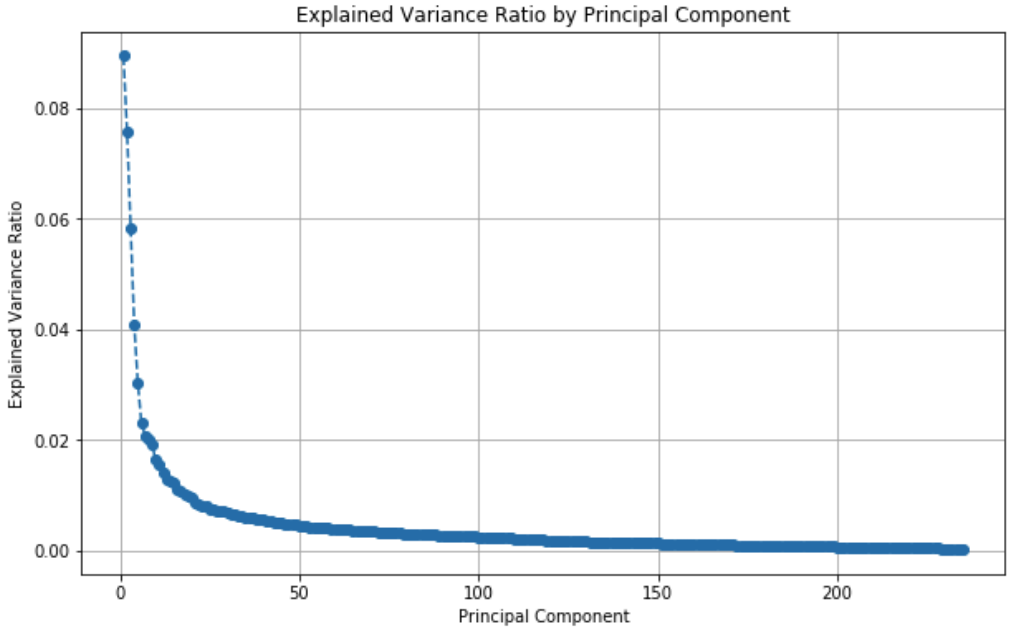
The approach taken to solve the problem of identifying potential customers spans across three main areas: Data Exploration, Unsupervised Learning and Supervised Learning Algorithms. During the data exploration, we clean and preprocess our data by removing attributes and records with missing data, unknown descriptions, invalid data types or high correlation to each other. During Unsupervised Learning, we reduce the dimensionality of the attributes by applying Principal Component Analysis, identify the optimal number of K-means clusters and compare how the customer dataset and general population differ from each other. Lastly, we apply Supervised Learning techniques with our training data to test different models for accuracy and ultimately pick the top performing model to fine-tune and optimize. For this exercise, we look at Logistic Regression, Random Forest Classification and Support Vector Classifier models. Random Forest Classification was deemed to perform best so we further optimize its performance by testing different parameters and then apply the completed model to the test set for making predictions.

The Expected Solution

Our expected solution is simple: create a way to accurately determine who in the population is most likely to become a customer and arm Arvato with the ability to acquire them through targeted marketing and acquisition tactics. Through our strategy, we are able to unveil the top characteristics of a person that determines the likelihood of them becoming a customer and develop a model that uses those characteristics to various degrees to make its prediction. Our solution allows a user to feed in a limitless dataset with the attributes our model has determined to be most explanatory and provides an output with those likelihoods of customer acquisition.

Metrics for Evaluation

We use a few different key metrics when evaluating the performance of our models. For unsupervised learning ,we look at the explained variance ratio by principal component after reducing dimensionality with PCA and we select the number of components that best explains the variance in outcomes. We aim for an explanation of at least 85% when making this selection and settle on a number of components equal to 100



With this number determined, we then calculate the optimal number of clusters to break our customers into for minimizing the squared sum of errors, which the sum of the squared differences between each observed data point and a corresponding predicted value. It's often used in the context of regression analysis, where you have a model that predicts values, and you want to measure how well the model fits the actual data.

$$SSE = \Sigma(y_i - \bar{y})^2$$

Where:

- Σ represents the summation symbol, which means to sum over all data points.
- y_i represents each observed data point.
- $\bar{y}$ represents the predicted value for each data point based on the regression model.

When it comes to supervised learning, the most important metrics are the accuracy and f-score of each model. With those in mind, we can determine each model's performance and pick the best one for further training and optimization. Accuracy measures the proportion of correctly predicted instances in a classification problem. It provides an overall assessment of a model's correctness.

Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

- True Positives (TP) are the number of correctly predicted positive instances.
- True Negatives (TN) are the number of correctly predicted negative instances.
- False Positives (FP) are the number of negative instances incorrectly predicted as positive.
- False Negatives (FN) are the number of positive instances incorrectly predicted as negative.

F-score is a metric that balances precision and recall. Precision measures how many of the predicted positive instances were actually positive, while recall measures how many of the actual positive instances were correctly

predicted. The F-score is the harmonic mean of precision and recall and provides a single metric for assessing a model's performance

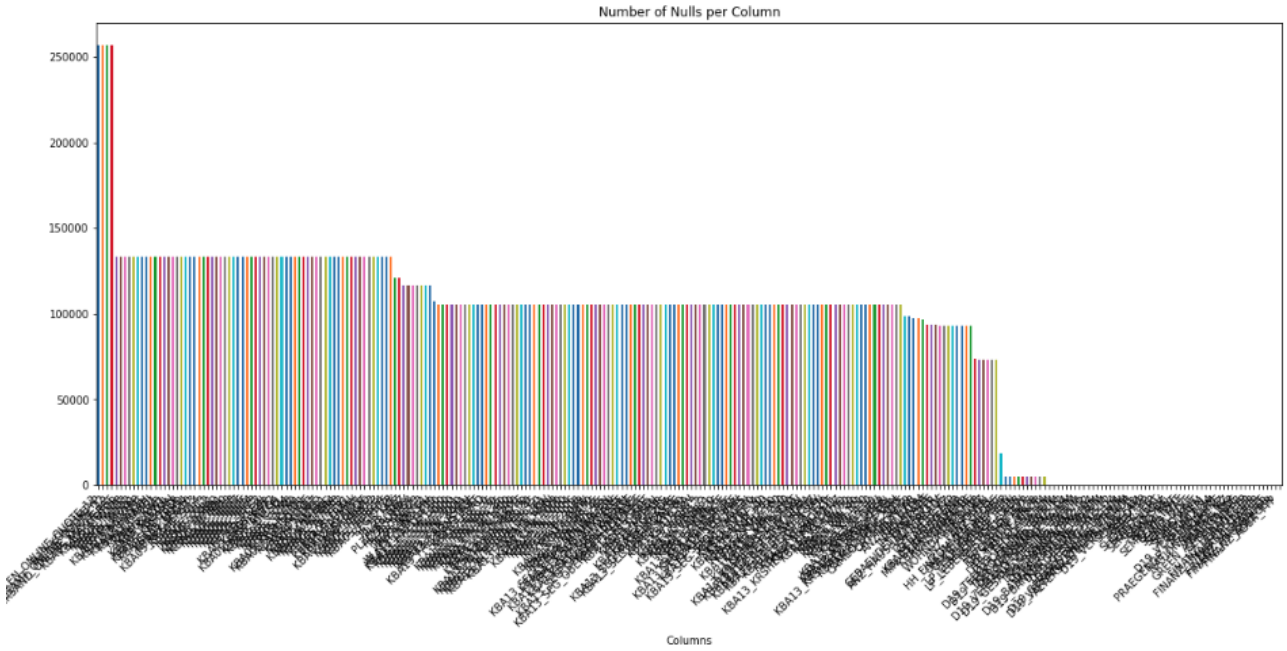
F-Score = 2 * (Precision * Recall) / (Precision + Recall)

- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)

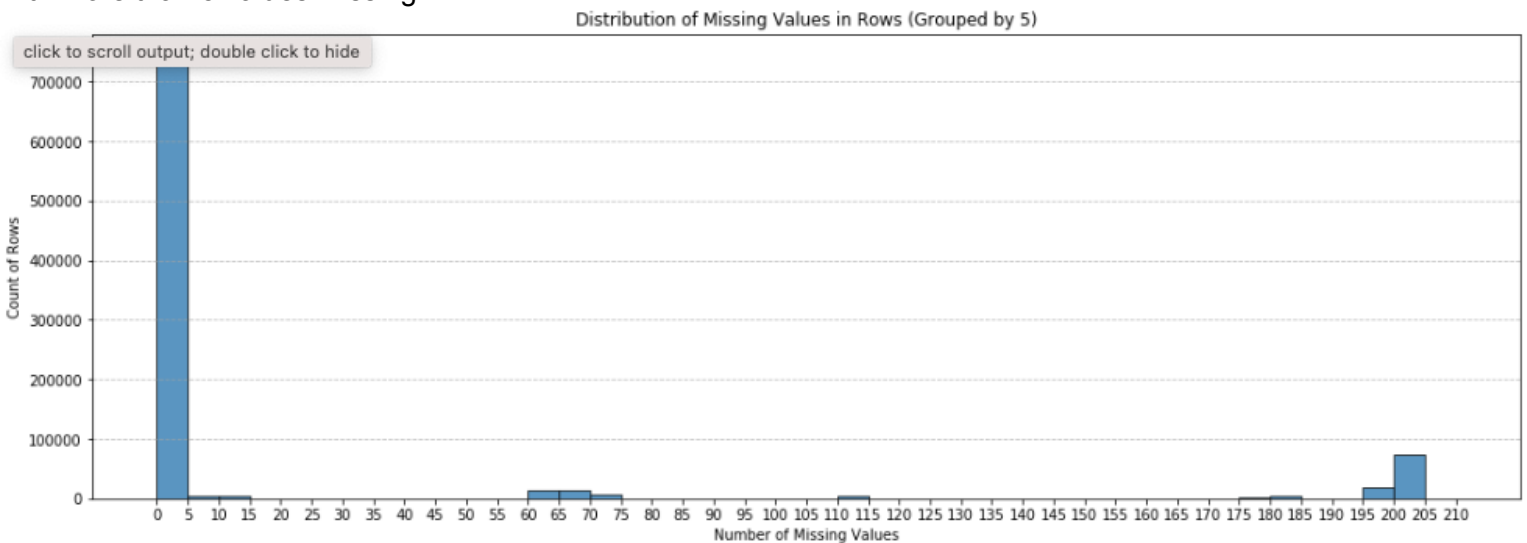
Data Exploration and Modeling Efforts

Exploratory Data Analysis

Diving deeper into our exploratory analysis, we uncovered many key insights that allowed us to eliminate unnecessary variables and enhance our data for improved modeling efforts later on. The first step is to eliminate any variables that we can't understand due to missing explanatory data about what the variable describes and how it is structured. This allows us to eliminate over 30 variables off the bat and then we can move on to step two. With so many records and columns, there were bound to be issues with the data and discrepancies that needed to be eliminated or addressed. The first piece of exploration deals with looking at the completeness of each attribute and what percent of records had a value for it. The below graph is busy but we can see that there are a number of columns with significantly more null values than others that can likely be eliminated once we see the percent makeup of the null values.



We determine that attributes with more than 28% of their records missing can be eliminated altogether, which allows us to go from 369 columns to 268 at this stage. Finally, we can drop any incomplete records, which we define as those with more than 5 values missing.



After this stage, we are able to go from 891,221 records to 743,322 and we can be confident that all of the columns and records retained are mostly complete and will allow us to make the most accurate model possible after a few more preprocessing steps.

Data Preprocessing

Now that the data has been cleaned up and those more complete records remain, we need to dive a bit deeper into the data types of each feature. For our modeling efforts to work, we only want numerical columns so we will need to either drop or re-encode any non-numerical values. Upon analyzing each variable, we identify 5 columns that need further review, 2 of which represent calendar years and 3 of which are object data types that need to be converted to numerical.

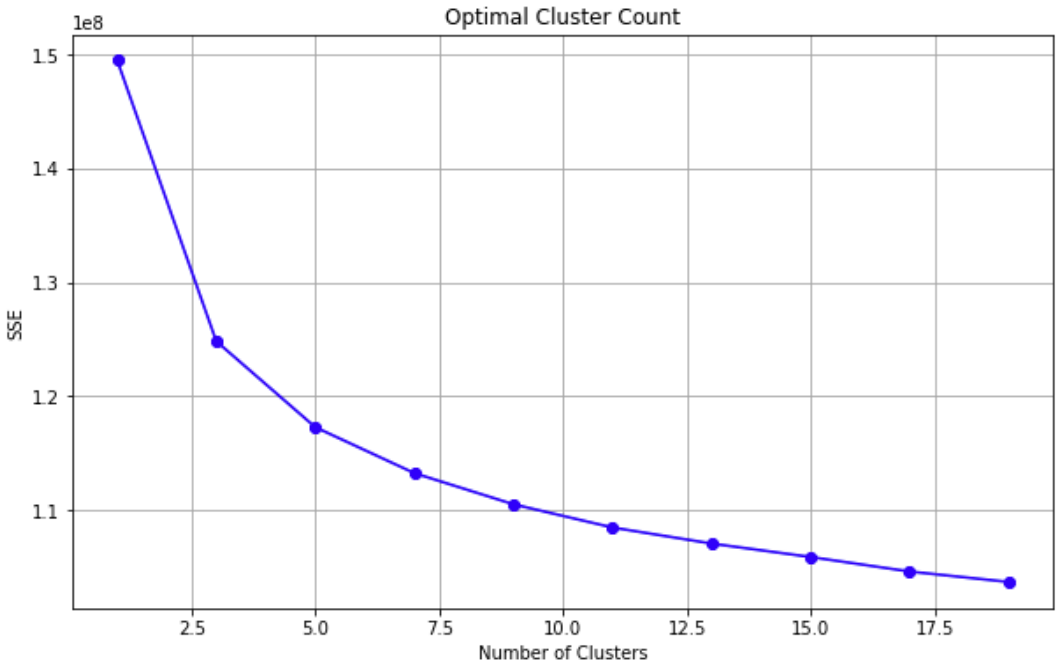
- CAMEO_DEU_2015: CAMEO classification 2015 - detailed classification
- CAMEO_DEUG_2015:CAMEO classification 2015 - Uppergroup
- OST_WEST_KZ: flag indicating the former GDR/FRG
- GEBURTSJAHR: year of birth
- MIN_GEBAEUDEJAHR: year the building was first mentioned in our database

It is decided that the 2 variables representing years and CAMEO_DEU_2015 will be dropped and the remaining 2 will be re-encoded, with OST_WEST_KZ having values of ‘W’ and ‘O’ that are converted to 1 and 0 and CAMEO_DEUG replacing non numeric values with 0 and converting all other numeric values to integers so they are uniform. Wrapping up, we now address potential correlation issues with all of our data in numerical format and ready to be analyzed with a correlation matrix. An additional 30 columns are identified as having correlations of over 85% with each other and removed from the dataset to avoid issues with multicollinearity. We can now impute any missing values in the dataset and we will be ready to move on to our modeling. It is identified that 27,044 of values are missing so we apply a forward and back fill imputation method to populate those records and then our data is complete and ready to be used for training.

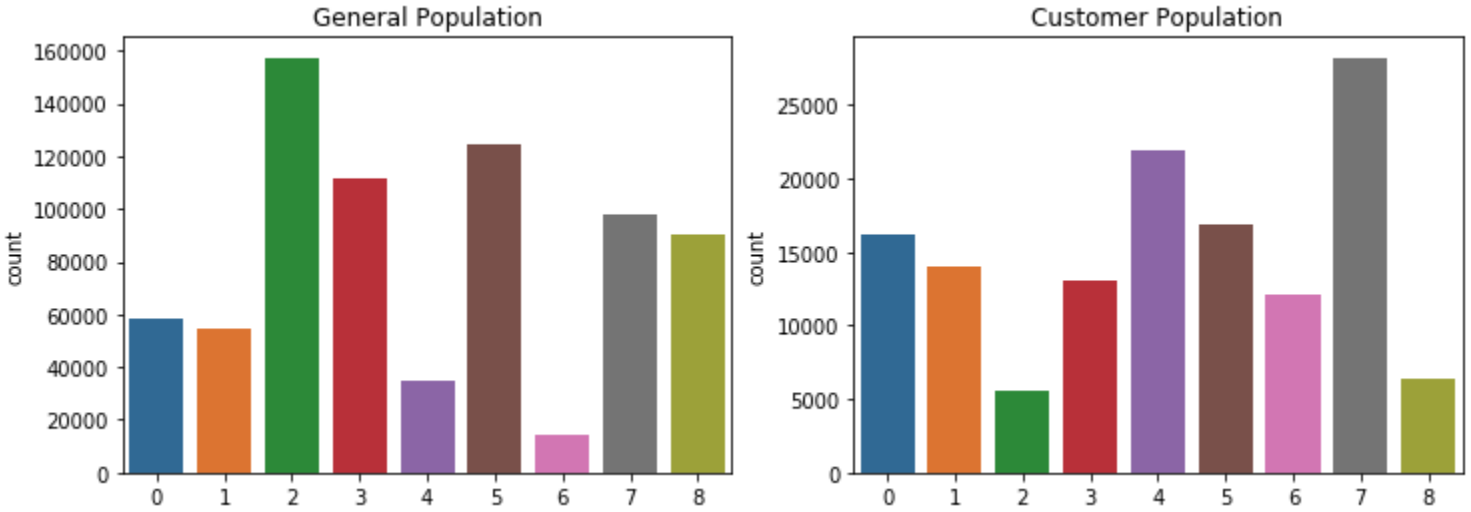
Unsupervised Learning

In this piece of the analysis, Principal Component Analysis (PCA) is applied to a dataset to reduce the data's dimensionality while retaining essential information, allowing us to explore the variance explained by each principal component. As discussed previously we first look at the explained variance ratio for all principal components to help us understand how much of the data's variance each component captures. We then narrow down to retaining a 100 principal components, which explains 85% of the variance in the general population and 90% in the customer population.. This choice is based on the observed explained variance ratio, with the goal is to capture a substantial portion of the variance while reducing dimensionality.

Next, the analysis moves into clustering using K-Means. The Within-Cluster Sum of Squares (WCSS) is calculated for varying numbers of clusters, ranging from 1 to 21. The WCSS represents the sum of squared distances of data points to their assigned cluster centers. By plotting the WCSS against the number of clusters, we aim to find an "elbow point," indicating an optimal cluster count. Upon identifying the optimal cluster count, which, in this case, appears to be 9 clusters, the K-Means model is re-fitted to the data. Cluster predictions are made for both the general population and the customer data.



To compare the customer and general population clusters, count plots are generated to visualize the proportion of data points in each cluster. These plots reveal how the customer population differs from the general population in terms of cluster distribution. This analysis helps uncover distinct patterns and groupings within the data, allowing for better segmentation and understanding of customer behavior. It's a crucial step in refining marketing strategies and enhancing customer acquisition by tailoring approaches to specific clusters or customer segments.



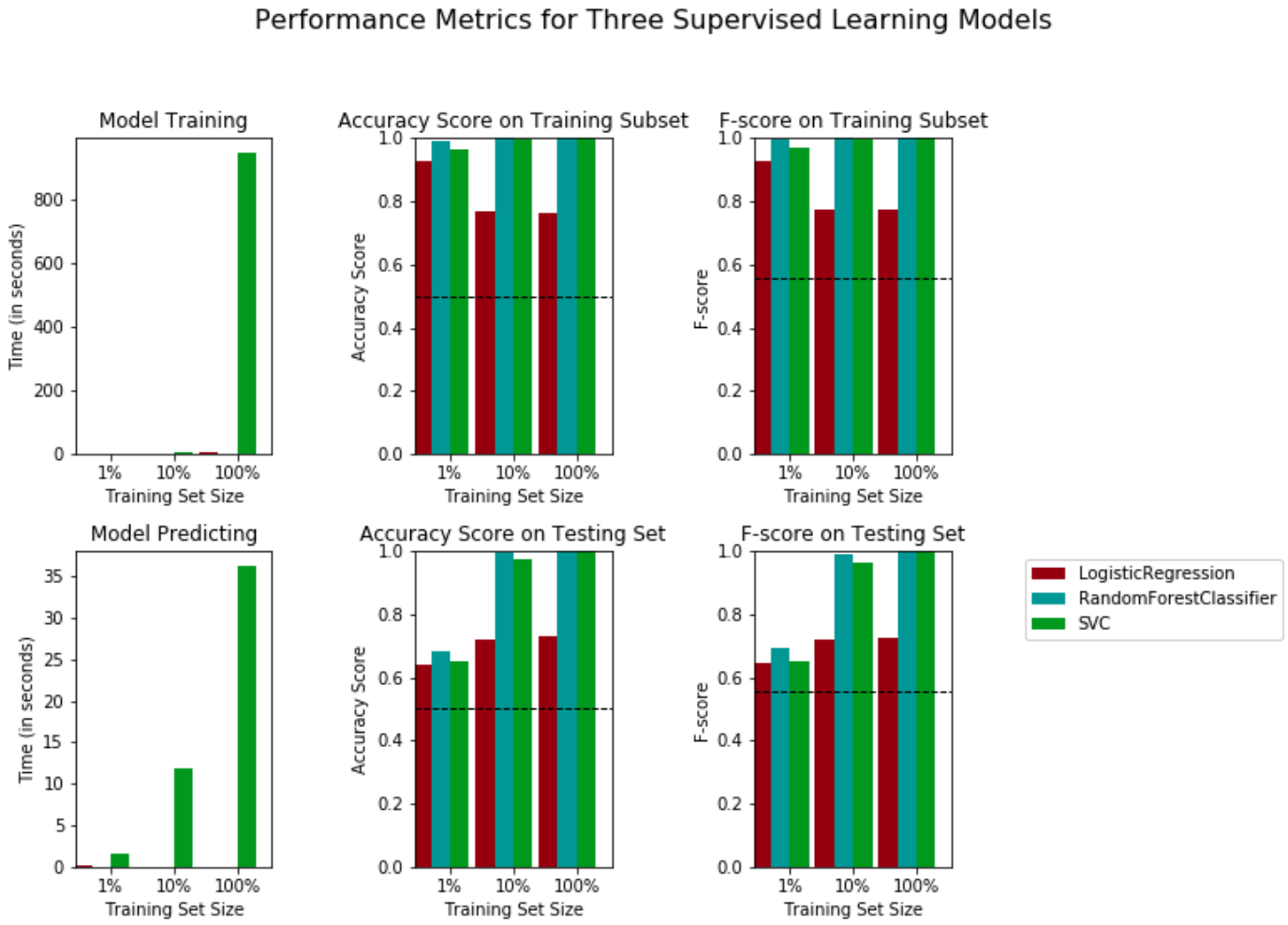
This is interesting because we are seeing a few major differences after clustering our data between the general population and customer population. Most notably, cluster_2 is severely underrepresented in the customer population, while cluster_4 and cluster_7 have significantly higher representation in the customer population.

Supervised Learning

During the supervised learning portion, loading in the new training data, ‘Udacity_MAILOUT_052018_TRAIN.csv’ and preparing it for modeling. We follow all the previous steps outlined for our Germany population and customer datasets where we clean and preprocess the data. Additionally, we also need to apply a resampling technique due to how skewed the responses are in our new training dataset. Resampling is needed to balance the response classes given that only 1% of the population had a response equal to 1 originally. This ensures that there is a sufficient representation of both positive (responding) and negative (non-responding) instances in the dataset, which is crucial for training and evaluating a model's performance.

Next, the features in the dataset are scaled using StandardScaler from scikit-learn. Scaling is essential to ensure that all features have the same influence on the machine learning models and to make computations more efficient. The dataset is then split into training and testing sets using train_test_split. This separation allows for model training and independent evaluation of the model's performance. A "naive predictor" is established, which serves as a baseline for evaluation. This predictor assumes that all individuals are classified as positive cases, and it calculates accuracy and F-score based on this assumption. These metrics provide a benchmark for evaluating more advanced models. Three machine learning models are selected for the analysis: Logistic Regression, Random Forest Classifier, and Support Vector Classifier (SVC). These models are initialized and trained on different proportions (1%, 10%, and 100%) of the training data to assess how they perform with varying sample sizes. Accuracy and F-scores are computed for each model on both the training and testing sets. These metrics provide insights into the models' predictive capabilities and show us that Random Forest Classifier is the best model to move forward with. It has a much higher accuracy than Logistic Regression and executes at a significantly faster pace than Support Vector Classifier.

Model Comparison



Hyperparameter Tuning

Finally, we can fine tune the hyperparameters to optimize the model further. GridSearchCV is used to search for the best combination of hyperparameters. In this case, we focus on the RandomForestClassifier parameters n_estimators, max_depth and min_samples_split:

- N_estimators: This parameter specifies the number of decision trees (estimators) that will be used in the random forest ensemble. The values provided in the list, such as 10, 50, and 100, represent different options for the number of trees to include in the forest. Increasing the number of trees can improve the model's robustness and accuracy, but it also increases computation time.
- Max_depth: The max_depth parameter controls the maximum depth of each decision tree in the forest. It can take values such as None (no maximum depth), 10, 20, and 30, which represent different choices for the maximum depth. A smaller maximum depth can help prevent overfitting, while a larger depth may allow the trees to capture more complex relationships in the data. The choice depends on the complexity of the problem and the amount of data.
- Min_samples_split: min_samples_split sets the minimum number of samples required to split an internal node during the construction of a decision tree. The provided values, such as 2, 5, and 10, represent different thresholds for the minimum number of samples. A smaller value can result in more splits, potentially capturing noise, while a larger value enforces a more general split criterion, which can help prevent overfitting.

Findings

Results

The results of all of this work are really exciting. We were able to predict with 100% accuracy and f-score which people are best positioned to be acquired for Arvato. We can be extremely confident that any people the model says are poised for converting to a customer will be acquirable. Specifically, when we run our model through a test set, we see that Of the 42,833 potential customers, 7,107 are more likely to be customers than not, 686 have a greater chance of being a customer, and 40 (38 + 2) are almost definitely going to be customers.

Conclusion

In our data-driven voyage, we started by delving into the depths of raw data, deciphering its enigmatic intricacies. Our quest led us through unsupervised learning, where we unveiled concealed patterns and transformed data chaos into crystal-clear insights. Moving into the realm of supervised learning, we meticulously refined our predictive model, fine-tuning it for peak accuracy and efficiency. Now, armed with this formidable tool, we stand ready to redefine our approach to customer acquisition and targeted marketing. The wisdom acquired at each leg of our journey acts as a guiding star, illuminating our strategic choices and charting a course toward a future brimming with triumphant endeavors.

Improvements

With how accurate the model is, I think we could use different criteria to better capture customers that are potential customers rather than hyper focusing on those that fit our criteria to a T and will without a doubt be customers. The over accuracy of the model can potentially hinder us in identifying customers that are more of a ‘maybe’ and rule them out so in this case we may want a wider net for who to target.

Acknowledgements

Through my prior experience with Udacity's Machine Learning nano-degree, I was able to leverage previous project work from my github to build out my supervised and unsupervised algorithms:
https://github.com/bchase17/machine_learning/tree/main