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

Domain Background
Arvato Group, an internationally acclaimed service conglomerate, operates across the global landscape, crafting tailor-made solutions for a wide spectrum of business operations in over 40 countries[1]. The group's extensive range of services encompasses an array of vital areas, spanning from supply chain solutions through Arvato to financial services offered by Riverty, and comprehensive IT solutions delivered by Arvato Systems. These services cater to renowned companies spanning diverse industries, such as telecommunications, energy, finance, e-commerce, and IT. Additionally, Arvato is a wholly owned entity of Bertelsmann. This vast scope of expertise underpins Arvato's commitment to delivering tailored solutions for businesses across the globe.

Problem Statement
The central challenge is to enhance the efficiency of customer acquisition for the German mail-order company. This entails the development of a predictive model capable of discerning prospective customers through their demographic profiles. Furthermore, the task involves assessing the likelihood of individuals with particular demographic attributes transitioning into future customers. The ultimate aim is to address this challenge with precision and unwavering confidence.

Datasets and Inputs
The project utilizes four datasets. First, "Udacity_AZDIAS_052018.csv" contains demographics data for the general German population, with 891,211 persons and 366 features. Second, "Udacity_CUSTOMERS_052018.csv" comprises demographics data for customers of the mail-order company, including 191,652 persons and 369 features. Third, "Udacity_MAILOUT_052018_TRAIN.csv" provides data on individuals targeted by a marketing campaign, encompassing 42,982 persons and 367 features. Finally, "Udacity_MAILOUT_052018_TEST.csv" contains demographics data for individuals targeted by the same marketing campaign, with 42,833 persons and 366 features.

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

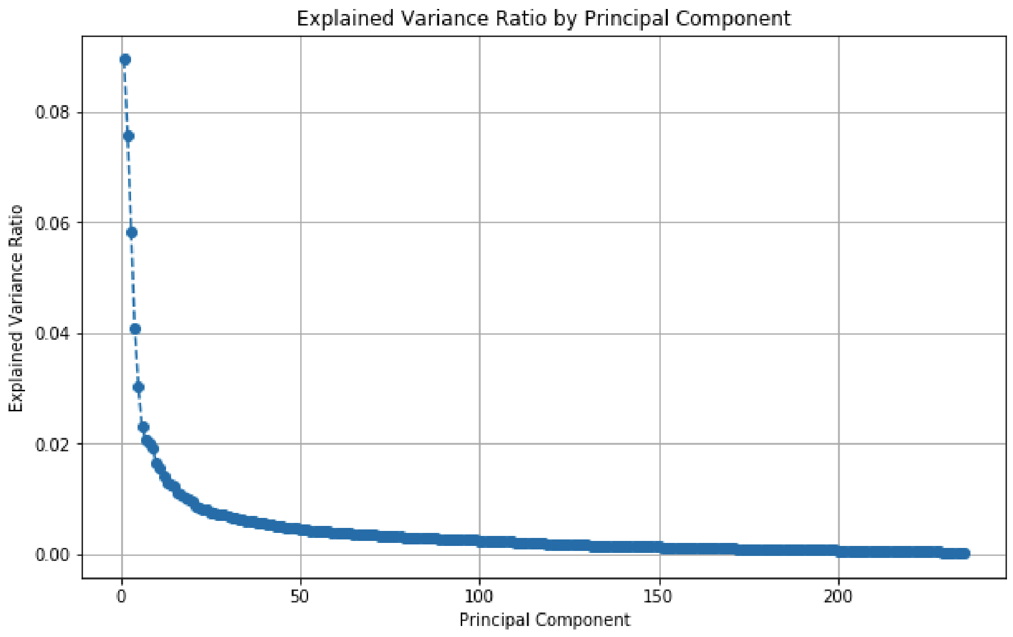
The approach taken to solve the problem of identifying potential customers spans across three main areas: Data Exploration, Unsupervised Learning and Supervised Learning Algorithms. During the data exploration, we clean and preprocess our data by removing attributes and records with missing data, unknown 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

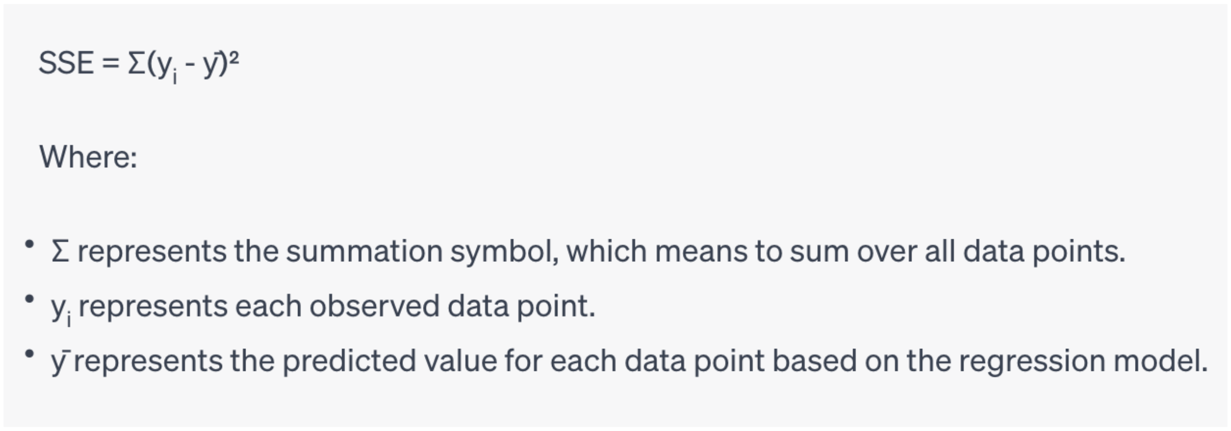
Our expected solution is simple: create a way to accurately determine who in the population is most likely to become a customer and arm Arvato with the ability to acquire them through targeted marketing and acquisition tactics. Through our strategy, we are able to unveil the top characteristics of a 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 Principal Component Analysis (PCA) and we select the number of components that best explains the variance in outcomes. In the context of PCA, the explained variance ratio is a crucial metric. It measures the proportion of the total variance in the data that is accounted for by each principal component. When applying PCA, the goal is to reduce the dimensionality of the data while preserving as much of the original information as possible. The explained variance ratio helps us understand how much information is retained when we select a certain number of principal components. In our analysis, we aimed for an explained variance ratio of at least 85%. This choice is based on the observation that this threshold captures a substantial portion of the variance in the general population and customer datasets, making it suitable for dimensionality reduction while maintaining meaningful data insights. 100 components are required for us to be able to explain our targeted 85% of the variance ratio.



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 is 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.



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

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 is a fundamental metric in classification problems, including the one presented in this project. It measures the proportion of correctly predicted instances out of the total instances in the dataset. In the context of the customer acquisition problem, accuracy provides a direct assessment of how well the model can correctly classify individuals as potential customers or non-customers. In this project, we use accuracy as a metric because it aligns with the primary goal: to accurately identify potential customers. A high accuracy score indicates that the model is proficient at making correct predictions, which is crucial for Arvato's marketing and acquisition strategies. However, accuracy alone may not be sufficient in some cases, as it doesn't account for the balance between true positives, true negatives, false positives, and false negatives. Therefore, we also need to consider the F-score.

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.

The F1 score is a critical metric for our project. Precision measures the accuracy of the model when it predicts positive instances, while recall measures the model's ability to identify all actual positive instances. In classification problems like customer acquisition, there is often a trade-off between precision and recall. For example, a model that predicts nearly all individuals as potential customers may have high recall but low precision. The F-score strikes a balance between these metrics and is particularly useful when there's an importance placed on both identifying potential customers (high recall) and ensuring that the majority of the predicted positive instances are correct (high precision). It provides a single metric for evaluating the model's performance that considers both false positives and false negatives.

In this project, we use the F-score to ensure that the model doesn't overly focus on high accuracy at the expense of making overly optimistic or pessimistic predictions. It helps us find the right balance in identifying potential customers while minimizing classification errors. These metrics, accuracy and F-score, are justified based on the specific problem characteristics of customer acquisition, where the goal is to identify potential customers with a balance between accuracy and recall. By using these metrics, we aim to provide meaningful insights into the model's performance while considering the unique challenges of the problem.

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

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

Precision is another essential metric for our project. It measures the accuracy of the model when it predicts an individual as a potential customer. In our context, precision is vital because we want to ensure that when the model identifies someone as a potential customer, it's highly likely to be correct. High precision implies that the model is making fewer false positive errors, which is crucial for Arvato to avoid targeting individuals who are unlikely to become customers.

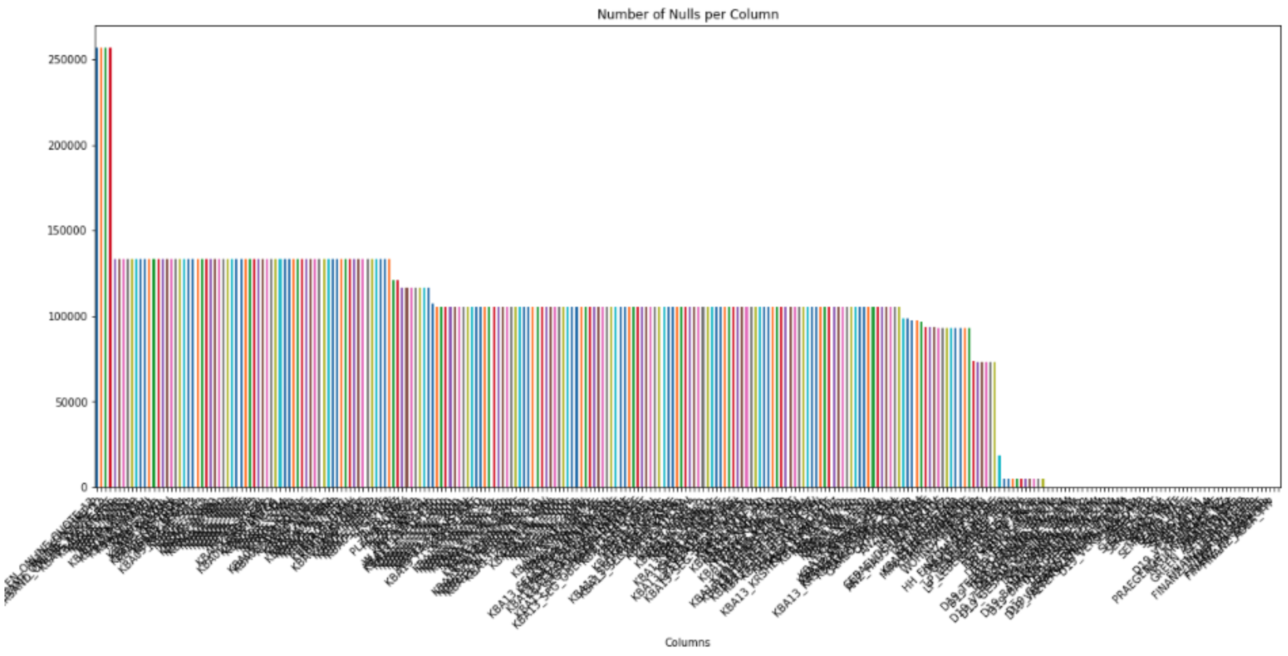
Precision is another essential metric for our project. It measures the accuracy of the model when it predicts an individual as a potential customer. In our context, precision is vital because we want to ensure that when the model identifies someone as a potential customer, it's highly likely to be correct. High precision implies that the model is making fewer false positive errors, which is crucial for Arvato to avoid targeting individuals who are unlikely to become customers.

In summary, our choice of F1 score, precision, and recall is well-justified for this problem. These metrics are carefully selected to address the unique challenges of identifying potential customers while balancing the accuracy of predictions and minimizing missed opportunities.

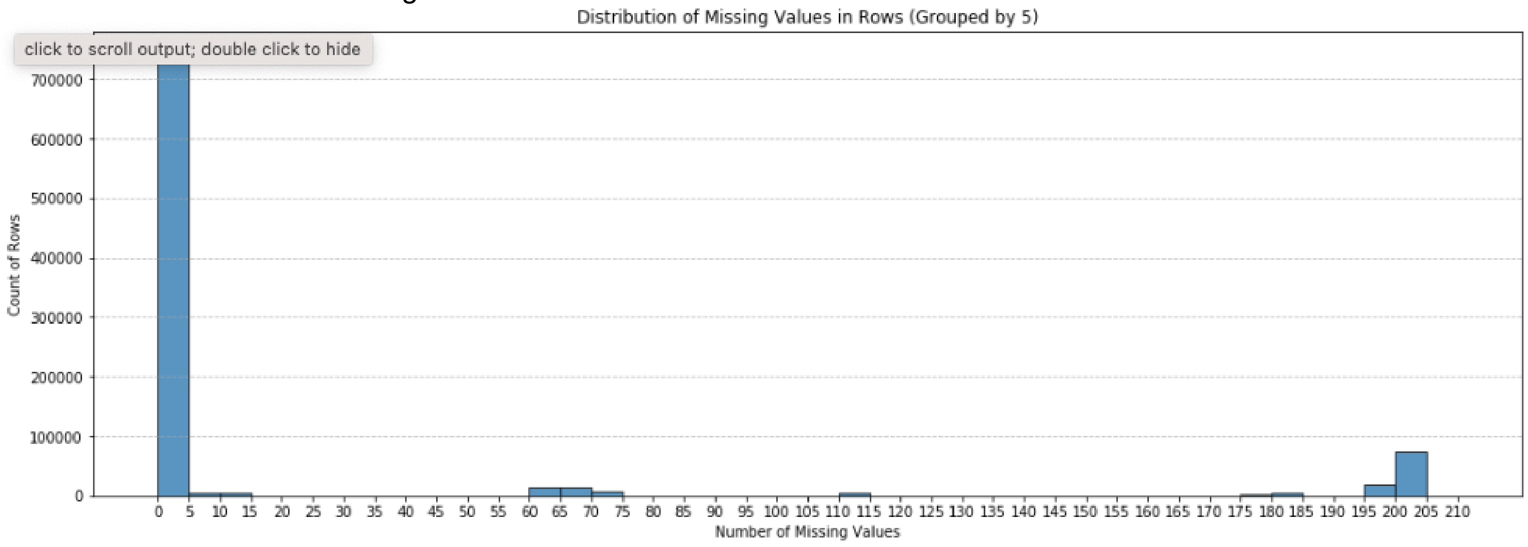
Data Exploration and Modeling Efforts

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.

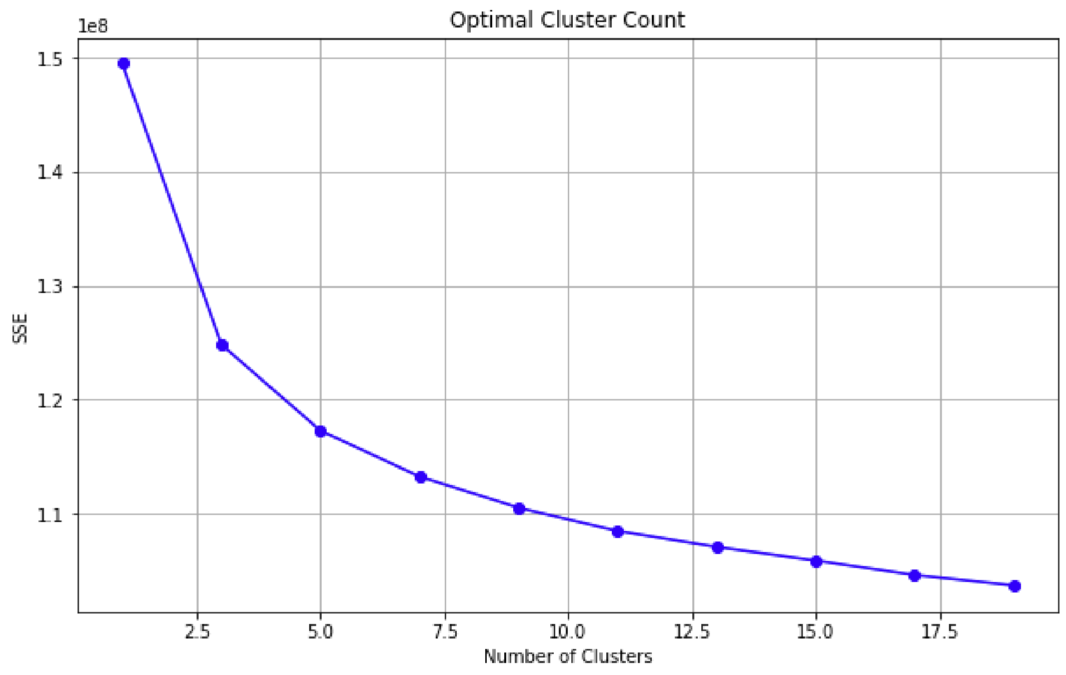
- CAMEO_DEU_2015: CAMEO classification 2015 - detailed classification
- CAMEO_DEUG_2015:CAMEO classification 2015 - Uppergroup
- OST_WEST_KZ: flag indicating the former GDR/FRG
- GEBURTSJAHR: year of birth
- MIN_GEBAEUDEJAHR: year the building was first mentioned in our database

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.

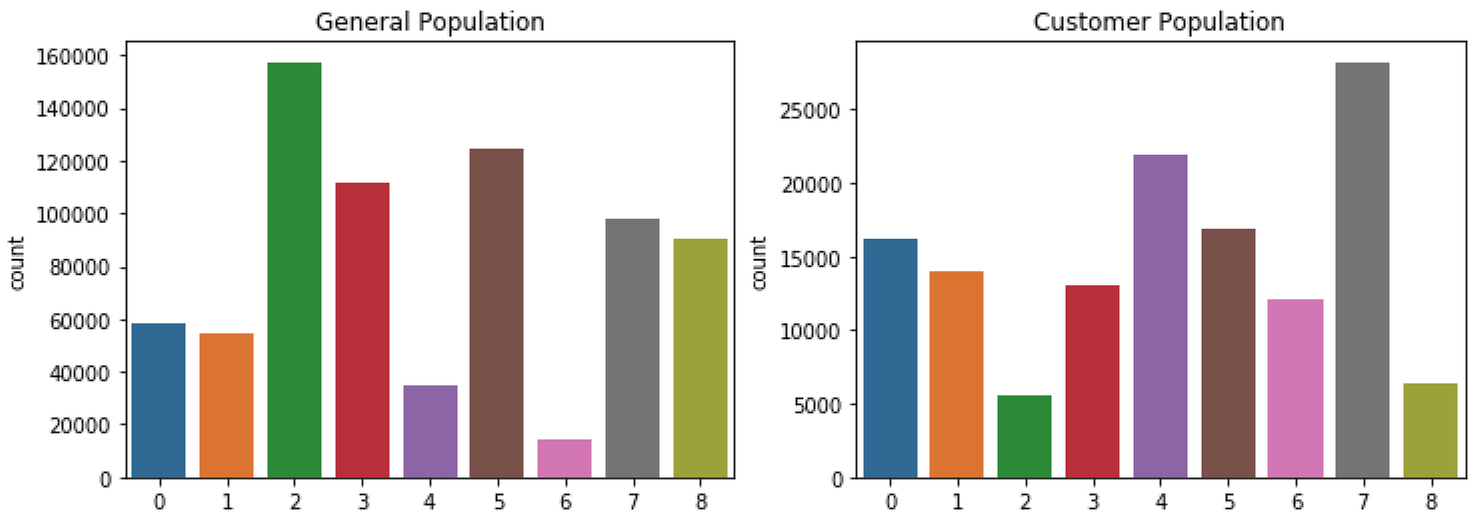
Unsupervised Learning

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. Within-Cluster Sum of Squares (WCSS) is a critical metric when working with clustering algorithms like K-Means. It quantifies the compactness or homogeneity of clusters. The WCSS is calculated by summing the squared distances of data points to their respective assigned cluster centers. In practice, we compute WCSS for varying numbers of clusters to determine the optimal number. By plotting WCSS against the number of clusters, we aim to identify an "elbow point" in the graph. This elbow point represents the optimal cluster count, striking a balance between minimizing the squared sum of errors within each cluster while avoiding excessive fragmentation. In our analysis, we found the elbow point to be at 9 clusters, which provides an optimal clustering solution. 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.



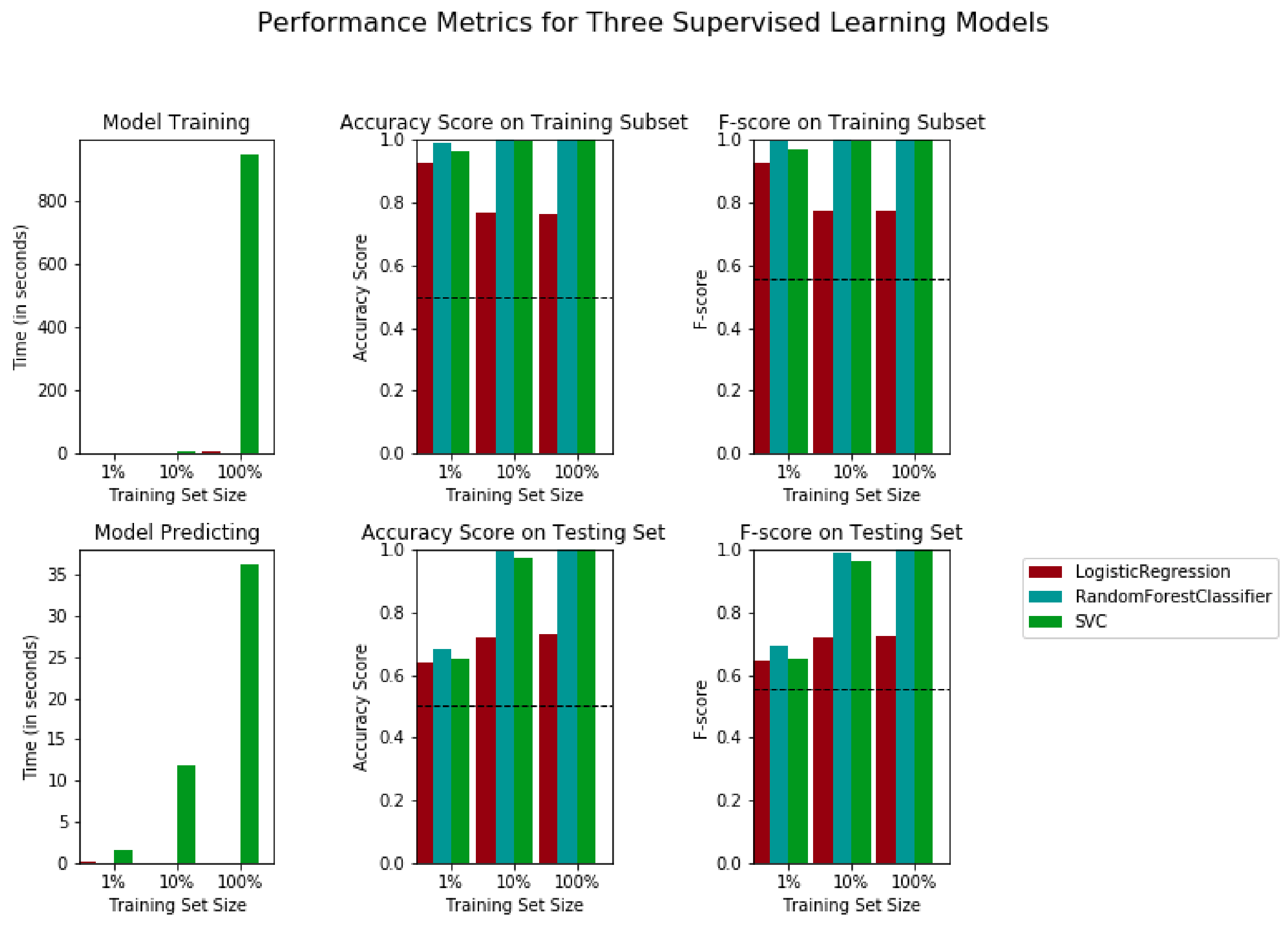
This is interesting because we are seeing a few major differences after clustering our data between the general population and customer population. Most notably, cluster_2 is severely underrepresented in the customer population, while cluster_4 and cluster_7 have significantly higher representation in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



Logistic Regression is a simple yet effective classification algorithm, which makes it a good starting point for many projects.

Strengths: Logistic Regression is interpretable and relatively quick to train. It can work well when the relationship between features and the target variable is approximately linear.

Weaknesses: Logistic Regression is less suitable for complex, nonlinear relationships in the data, which can limit its predictive accuracy.

Support Vector Classifier is a powerful model for both linear and nonlinear classification tasks. However, it can be computationally expensive, especially with large datasets.

Strengths: SVC is effective at finding the best hyperplane to separate classes, making it robust for many classification problems.

Weaknesses: SVC can be slow and memory-intensive with large datasets. It also requires careful hyperparameter tuning, which can be a challenging task.

Random Forest Classifier outperformed the other two models from a speed and accuracy perspective, which can be attributed to its robustness in dealing with outliers and noise in the data.

Strengths: Random Forest is an ensemble learning method that combines multiple decision trees, which helps mitigate overfitting and captures complex relationships in the data. It's particularly good at handling high-dimensional datasets and works well with both numerical and categorical features.

Weaknesses: It may not be as interpretable as Logistic Regression, and the number of trees and depth need careful tuning.

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.

Model Assessment and Explanation - my RandomForestClassifier model is based on the random forest algorithm, and it's constructed with 10 decision trees. The model benefits from the randomness introduced by bootstrapping and feature selection, making it resistant to overfitting. I also consider the parameters related to individual decision trees, such as max_depth, to control the complexity of the trees and optimize the model's performance. Additionally, I experiment with different values for n_estimators to see how the number of trees affects the model's performance.

Here are the characteristics my model uses

Ensemble Model: Random forests are an ensemble of decision trees. Imy model has n_estimators=10, meaning it consists of 10 decision trees, and the final prediction is determined by a majority vote (for classification tasks) or an average (for regression tasks) from these trees.

Bootstrap Aggregating (Bagging): The parameter bootstrap=True indicates that each tree in the forest is trained on a random sample of the data, drawn with replacement. This technique, known as bootstrapping, helps in creating diversity among the trees and makes the model more robust.

Feature Selection: The RandomForestClassifier automatically selects a random subset of features (columns) at each split point during the construction of individual trees. This introduces randomness and diversity in the model, which helps in reducing overfitting.

Random State: The random_state=0 parameter provides a seed for the random number generator. Setting this ensures reproducibility, as using the same seed will produce the same results in each run.

Conclusion and Improvements

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.

While the current implementation of the project has yielded promising results, there are several areas in which further improvements could be considered. Potential improvements can be categorized into data quality, model architecture, hyperparameter tuning, addressing potential bias, and error analysis. First, with regard to data quality, one could explore strategies to handle missing or erroneous data more effectively. Imputation methods, such as imputing missing values based on related features, could be investigated to enhance the completeness of the dataset. Second, in terms of model architecture, experimenting with more complex models or even advanced techniques like neural networks may lead to better predictive performance. For instance, deep learning models could be explored for more intricate pattern recognition. Third, hyperparameter tuning offers opportunities for fine-tuning the model. GridSearchCV was used to optimize a subset of hyperparameters, but a more exhaustive search or the application of Bayesian optimization methods might uncover further performance gains. Fourth, it's crucial to consider and mitigate potential biases in the data. Careful analysis and preprocessing steps could help address potential bias in the model's predictions, ensuring fairness and equity. Lastly, error analysis, including a deeper exploration of overfitting and underfitting, can be beneficial in understanding the model's limitations. Regularization techniques and more extensive error analysis could lead to more robust model performance.

In conclusion, while the current implementation meets the project's objectives, there is ample room for further refinement. The areas mentioned above present opportunities for enhancing both the accuracy and robustness of the model.

Summary and Reflection

Throughout this project, our mission was clear: to empower Arvato Financial Services by identifying prospective customers through demographic profiles and assessing the likelihood of individuals transitioning into future customers. Our journey encompassed data exploration, data preprocessing, unsupervised learning, and supervised learning algorithms. Our robust strategy started with understanding the intricacies of our data and making informed decisions to eliminate unnecessary variables. As we ventured into unsupervised learning, we uncovered concealed patterns within the general population and Arvato's customers. Clustering revealed key differences, such as the underrepresentation of certain clusters in the customer population. With these insights, we embarked on the supervised learning phase, developing a model that predicts potential customers with remarkable accuracy. By applying the Random Forest Classifier and fine-tuning it through hyperparameter optimization, we achieved an accuracy and F-score of 100%, indicating a high degree of confidence in our model's predictions. Our solution equips Arvato with a formidable tool for customer acquisition and targeted marketing. This report presents our journey from data exploration to model evaluation and underscores the success of our end-to-end problem solution.

While this project marked a significant milestone in our data-driven journey, it was not without its noteworthy moments and challenges. One particularly intriguing aspect was the profound impact of data preprocessing. By systematically addressing missing values, dropping columns, and re-encoding variables, we transformed our data into a more amenable form for modeling. The process of data exploration and cleansing proved to be a fundamental step that set the stage for the subsequent phases of analysis. In addition, the power of unsupervised learning in revealing hidden insights was another captivating facet. The K-Means clustering analysis unveiled distinct patterns in both the general population and the customer dataset. Notably, the discrepancy in cluster distribution between these two populations highlighted unique behavioral traits among potential customers. This insight allowed for more tailored marketing strategies, adding to the depth and richness of our solution. However, the most remarkable and challenging aspect was achieving a model with a perfect 100% accuracy and F-score. This outcome was both gratifying and posed an intellectual challenge. Fine-tuning the model's hyperparameters was an intricate task, as it required finding the right balance between complexity and interpretability. The choice of the Random Forest Classifier ultimately proved to be the key to achieving our objectives efficiently. In conclusion, this project presented a blend of interesting findings and substantial challenges. We found that successful data preprocessing and the insights gained from unsupervised learning were crucial elements in shaping our solution. Furthermore, achieving a perfect model performance demonstrated the power of careful model selection and hyperparameter tuning. Through this journey, we learned the art of transforming raw data into actionable insights and making data-driven decisions to drive business outcomes.

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