Project Report

Problem Statement:

In this capstone project, I will analyze demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population. I will then identify the parts of the population that best describe the core customer base of the company. Then, I will apply what I've learned on a third dataset with demographics information for targets of a marketing campaign for the company, and use a model to predict which individuals are most likely to convert into becoming customers for the company

Datasets and Inputs:

There are 3 data files that will be used in this project:

- **`Udacity_AZDIAS_052018.csv`**: Demographics data for the general population of Germany
- **`Udacity_CUSTOMERS_052018.csv`**: Demographics data for customers of a mail-order company
- **`Udacity_MAILOUT_052018_TRAIN.csv`**: Demographics data for individuals who were targets of a marketing campaign
- **`Udacity_MAILOUT_052018_TEST.csv`:** Demographics data for individuals who were targets of a marketing campaign

Benchmark Model:

Since the problem can be modeled as a binary classification problem where 1 = converted to customer and 0 = not converted to customer, I propose using a logistic regression model as a benchmark model as this is a common, simple and time-efficient way to benchmark classification problems.

Evaluation Metrics:

I propose using F1-score as an evaluation metric as we are dealing with a binary classification problem and since we are looking at an imbalance dataset, a metric like the F1 score would

paint a better picture of the model's efficacy than a simple metric like accuracy or precision and recall.

$$F1\ Score\ =\ 2\ *\ \frac{Precision * Recall}{Precision + Recall}$$

Project Development In-Detail:

Prepare the Data (Understanding the data + EDA)

In part 0.1 of the notebook, I used various functions that I learned throughout my nanodegree program to see what the data looks like (using head(), size(), info() etc.) and understand the nature of the data.

What I noticed is that a large number of columns had NaN/Missing values that could possibly affect the performance of the model later. Columns like **ALTER_KIND1** , **ALTER_KIND2,** **ALTER_KIND3** and **ALTER_KIND4** had most of its data missing.

ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN

I also noticed that **'KK_KUNDENTYP', 'EXTSEL992'** also had a large number of missing values thus these columns were discarded in the data cleaning phase

	CAMEO_DEU_2015	CAMEO_DEUG_2015	CAMEO_INTL_2015	D19_LETZTER_KAUF_BRANCHE	EINGEFUEGT_AM	OST_WEST_KZ	PRODUCT_GROUP
0	1A	1	13	D19_UNBEKANNT	1992-02-12 00:00:00	W	COSMETIC_AND_FOOD
1	NaN	NaN	NaN	D19_BANKEN_GROSS	NaN	NaN	FOOD
2	5D	5	34	D19_UNBEKANNT	1992-02-10 00:00:00	W	COSMETIC_AND_FOOD
3	4C	4	24	D19_NAHRUNGSERGAENZUNG	1992-02-10 00:00:00	W	COSMETIC
4	7B	7	41	D19_SCHUHE	1992-02-12 00:00:00	W	FOOD

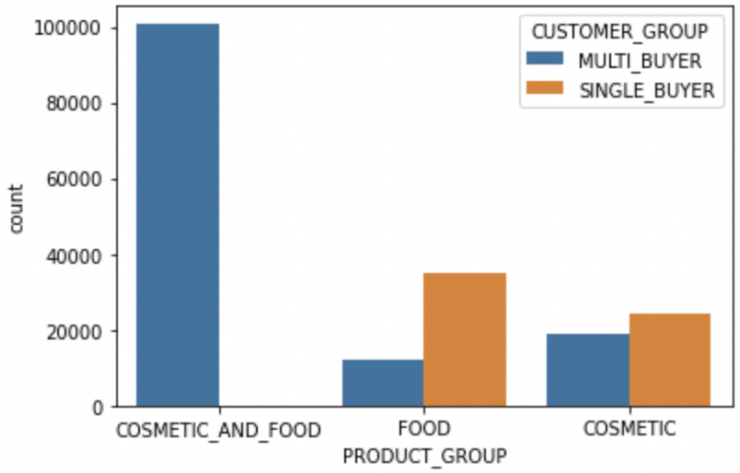
As can be seen from the above table, the following columns had categorical values and had to be processed accordingly

- **'CAMEO_DEUG_2015'** and **'CAMEO_INTL_2015'** are categorical variables that map categories to numbers
- **'EINGEFEUGT_AM'** is a date variable that will be processed to datetime in later stages
- **'OST_WEST_KZ'** is a binary category with 'W' representing West and 'O' representing east
- **'CAMEO_DEUG_2015'** and **'CAMEO_INTL_2015'** also seems to have XX values that I have assumed to be an anomaly and have ignored since there is very few occurrences

The customers dataset has an additional 3 columns that is not present in the population dataset as shown below

	CUSTOMER_GROUP	ONLINE_PURCHASE	PRODUCT_GROUP
0	MULTI_BUYER	0	COSMETIC_AND_FOOD
1	SINGLE_BUYER	0	FOOD
2	MULTI_BUYER	0	COSMETIC_AND_FOOD
3	MULTI_BUYER	0	COSMETIC
4	MULTI_BUYER	0	FOOD

What can be seen from the above screenshot and the table below is that the cosmetic and food group are only multi buyers while the food group has substantially more single buyers than multi buyers. Additionally the two groups are somewhat equal in the cosmetic category.



Feature Engineering

- Additional features are created from the earlier **'EINGEFEUGT_AM'** column which is converted to datetime. We then glean day, month and year features from it

```
def additional_features(df, date_col):
    df[date_col] = pd.to_datetime(df[date_col])

    df[date_col+"_Day"] = df[date_col].dt.day
    df[date_col+"_Month"] = df[date_col].dt.month
    df[date_col+"_Year"] = df[date_col].dt.year

    df = df.drop(date_col, axis=1)

    return df
```

- I then map the binary categories ‘W’ and ‘O’ into 1 and 0 for easier understanding of the model

```
azdias_df['OST_WEST_KZ'] = azdias_df['OST_WEST_KZ'].map({'W': 1, 'O': 1})
customers_df['OST_WEST_KZ'] = customers_df['OST_WEST_KZ'].map({'W': 1, 'O': 1})
```

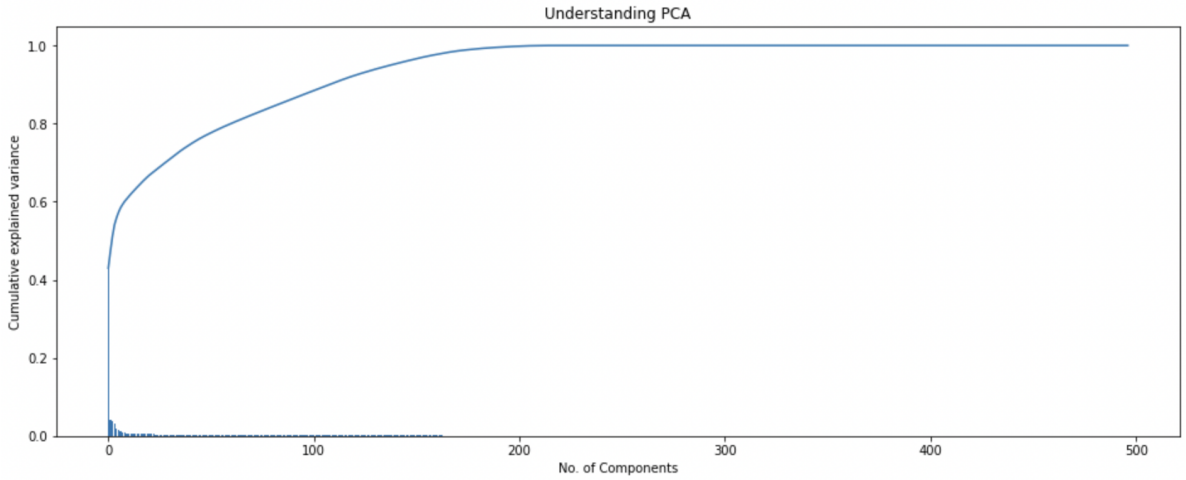
- I then used one-hot encoding to encode the categorical features in the data

```
    dummies = pd.get_dummies(df[cat_cols])
    df = pd.concat([df, dummies], axis=1)
    df = df.drop(cat_cols, axis=1)

    return df
```

- After this I scaled the features so they all fit in a standard range and the model produces more accurate results

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
azdias_scaled_df = sc.fit_transform(azdias_df)
customers_scaled_df = sc.transform(customers_df)
```

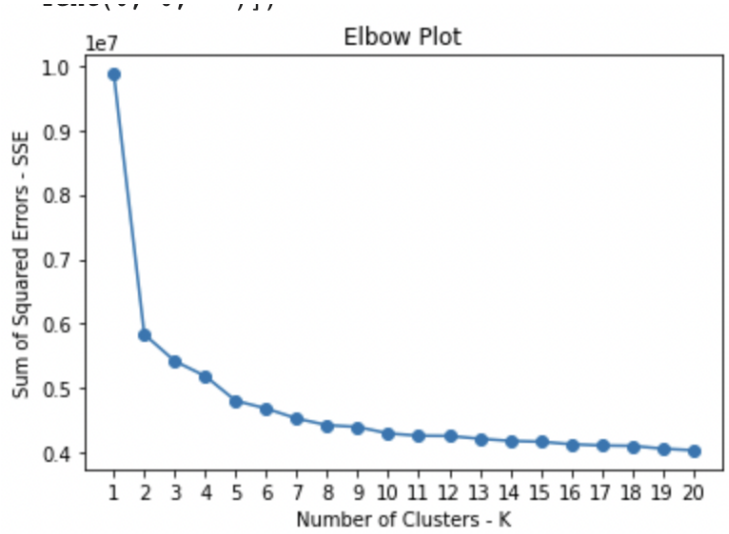
- Additionally, I performed PCA since we have a large volume of data so that it can be visualized and digested by the model easily



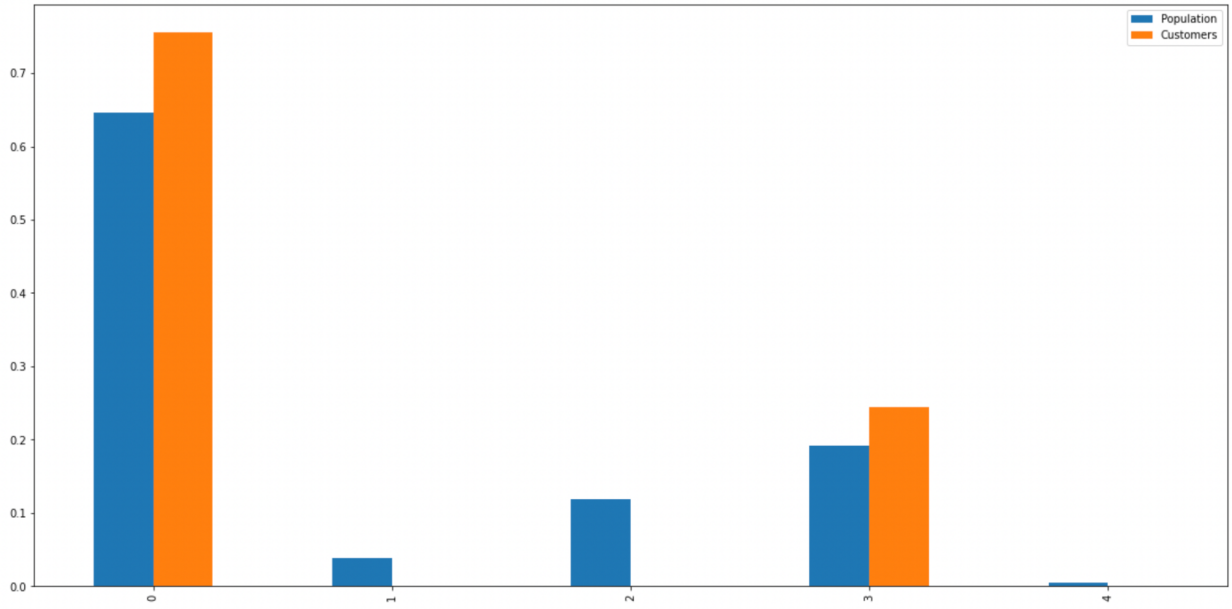
This graph shows us that close to 200 features explains most of the variance as opposed to the ~500 features present in the data.

Create the customer segmentation report:

We segment the customers into groups using K-Means. To understand the optimal number of clusters, I used the elbow method as shown below



This plot shows K=2 to be the optimal no. of clusters, however I chose K=5 to get better segmentation with more than a binary cluster. The clustering model is trained on K=5 and use this model to predict labels for the customer and population datasets



It can be seen that clusters 0 and 3 have similarities between the customer and population datasets. This is useful because we can then assume that these clusters can better help us find new customers in the general population.

Build prediction model:

I first started by building and training the baseline model mentioned in my proposal - the Logistic Regression model. Once this was done, I trained a decision tree, random forest and gradient classifier model. But all of their F1 scores were 0 or close to 0. To find out what the issue was by seeing the target response in the population dataset.

```
y.value_counts()
✓ 0.2s

0    42430
1      532
Name: RESPONSE, dtype: int64
```

This shows us that the data is highly imbalanced with a very small proportion of response values being 1 as opposed to 0. I performed upsampling on the data that provided a response of 1 to balance the data and then trained the models again. This time there was an improvement in the performance of both the baseline and the best performing model - The Gradient Boosting Classifier

Model	F1 score	ROC AUC
Logistic Regression	0.03763	0.60249
Gradient Boosting	0.06715	0.68998

References:

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html
https://towardsdatascience.com/elbow-method-is-not-sufficient-to-find-best-k-in-k-means-clustering-fc820da0631d#:~:text=The%20elbow%20method%20is%20a,cluster%20and%20the%20cluster%20centroid.
https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c