Machine Learning Engineer Nanodegree

Capstone Project

Joe Abraham
September 20, 2020

Definition

The objective of this project is to analyze demographics data of customers of a mail-order sales company in Germany that sells organic products and compare that to demographics data of the general population of Germany. The end goal is to be able to predict, based on demographics data, which individuals from the general population should be targeted in the mail-order campaign. Both unsupervised and supervised learning techniques will be used. Unsupervised learning will be used to help identify segments of the general population of Germany that best matches the existing customer base of the company. A supervised learning prediction model will be developed to predict the likelihood of whether or not an individual of the general population will become a customer. The dataset was provided by a real business - Bertelsmann Arvato Analytics and represents a real-life data science task.

The whole project is divided into 4 parts:

- Part 0 : Know the data, where you try to understand the data
- Part 1 : Customer Segmentation (Unsupervised Learning)
- Part 2 : Supervised Learning
- Part 3 : Kaggle competition

About Arvato

Arvato is an internationally active services company that develops and implements innovative solutions for business customers from around the world. These include SCM solutions, financial services and IT services, which are continuously developed with a focus on innovations in automation and data/analytics. Globally renowned companies from a wide variety of industries – from telecommunications providers and energy providers to banks and insurance companies, e-commerce, IT and Internet providers – rely on Arvato's portfolio of solutions.

Arvato is wholly owned by Bertelsmann. The services business also includes the Majorel group of companies, in which Bertelsmann owns 50 percent of shares.

About Dataset

There are four data files associated with this project:

- Udacity_AZDIAS_052018.csv: Demographics data for the general population of Germany; 891 211 persons (rows) x 366 features (columns).
- Udacity_CUSTOMERS_052018.csv: Demographics data for customers of a mail-order company; 191 652 persons (rows) x 369 features (columns).
- Udacity_MAILOUT_052018_TRAIN.csv: Demographics data for individuals who were targets of a marketing campaign; 42 982 persons (rows) x 367 (columns).
- Udacity_MAILOUT_052018_TEST.csv: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

Due to privacy concern - Datasets are protected and is not publicly available

Evaluation Metrics

The evaluation metric for this competition is AUC for the ROC curve, relative to the detection of customers from the mail campaign. A ROC, or receiver operating characteristic, is a graphic used to plot the true positive rate (TPR, proportion of actual customers that are labeled as so) against the false positive rate (FPR, proportion of non-customers labeled as customers).

The line plotted on these axes depicts the performance of an algorithm as we sweep across the entire output value range. We start by accepting no individuals as customers (thus giving a 0.0 TPR and FPR) then gradually increase the threshold for accepting customers until all individuals are accepted (thus giving a 1.0 TPR and FPR). The AUC, or area under the curve, summarizes the performance of the model. If a model does not discriminate between classes at all, its curve should be approximately a diagonal line from (0, 0) to (1, 1), earning a score of 0.5. A model that identifies most of the customers first, before starting to make errors, will see its curve start with a steep upward slope towards the upper-left corner before making a shallow slope towards the upper-right. The maximum score possible is 1.0, if all customers are perfectly captured by the model first. (It should be noted that this particular task is very difficult with a lot of noise, and so you should not expect extremely high scores!)

Data Exploration, Preprocessing and Cleaning

Data Exploration

How a sample of general population dataset looks like

In [41]:
```
display(azdias.head())
```

	Unnamed: 0	LNR	AGER_TYP	AKT_DAT_KL	ALTER_HH	ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4	ALTERSKATEGORIE_FEIN	ANZ_HA
0	0	910215	-1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	910220	-1	9.0	0.0	NaN	NaN	NaN	NaN	21.0	11.0
2	2	910225	-1	9.0	17.0	NaN	NaN	NaN	NaN	17.0	10.0
3	3	910226	2	1.0	13.0	NaN	NaN	NaN	NaN	13.0	1.0
4	4	910241	-1	1.0	20.0	NaN	NaN	NaN	NaN	14.0	3.0

5 rows × 367 columns

How a sample of customer dataset looks like

In [41]:
```
display(azdias.head())
```

	Unnamed: 0	LNR	AGER_TYP	AKT_DAT_KL	ALTER_HH	ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4	ALTERSKATEGORIE_FEIN	ANZ_HA
0	0	910215	-1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	910220	-1	9.0	0.0	NaN	NaN	NaN	NaN	21.0	11.0
2	2	910225	-1	9.0	17.0	NaN	NaN	NaN	NaN	17.0	10.0
3	3	910226	2	1.0	13.0	NaN	NaN	NaN	NaN	13.0	1.0
4	4	910241	-1	1.0	20.0	NaN	NaN	NaN	NaN	14.0	3.0

5 rows × 367 columns

How unknown values are represented in each column

Attribute	Value	Meaning
AGER_TYP	-1	unknown
ALTERSKATEGORIE_GROB	-1, 0	unknown
ALTER_HH	0	unknown / no main age detectable
ANREDE_KZ	-1, 0	unknown
BALLRAUM	-1	unknown
BIP_FLAG	-1	unknown
CAMEO_DEUG_2015	-1	unknown
CAMEO_DEUINTL_2015	-1	unknown
CJT_GESAMTTYP	0	unknown
D19_KK_KUNDENTYP	-1	unknown
EWDICHTE	-1	unknown
FINANZTYP	-1	unknown
FINANZ_ANLEGER	-1	unknown
FINANZ_HAUSBAUER	-1	unknown
FINANZ_MINIMALIST	-1	unknown
FINANZ_SPARER	-1	unknown
FINANZ_UNAUFFAELLIGER	-1	unknown
FINANZ_VORSORGER	-1	unknown
GEBAEUDETYP	-1, 0	unknown
GEOSCORE_KLS7	-1, 0	unknown
HAUSHALTSSTRUKTUR	-1, 0	unknown
HEALTH_TYP	-1	unknown
HH_EINKOMMEN_SCORE	-1, 0	unknown
INNENSTADT	-1	unknown
KBA05_ALTER1	-1, 9	unknown
KBA05_ALTER2	-1, 9	unknown
KBA05_ALTER3	-1, 9	unknown
KBA05_ALTER4	-1, 9	unknown
KBA05_ANHANG	-1, 9	unknown
KBA05_ANTG1	-1	unknown
KBA05_ANTG2	-1	unknown
KBA05_ANTG3	-1	unknown
KBA05_ANTG4	-1	unknown
	-1, 9	unknown
KBA05_BAUMAX	-1, 0	unknown
KBA05_CCM1	-1, 9	unknown

Data Pre-Processing and Cleaning

After data exploration I came up with a function to clean the dataset

```
def data_cleaning(df,clustering=False):
    '''
    Replace strings
    Replace missing values to np.nan - data taken from the excel provided.
    Replace na with mean
    LNR set as index
    Removal of outliers
    Cleaning columns if it doesn't have enough data
    '''
    print(df.shape)

    for column in ['CAMEO_DEUG_2015', 'CAMEO_DEU_2015', 'CAMEO_INTL_2015']:
        try:
            df[column] = df[column].replace(["X","XX"],np.nan)
        except:
            pass

    for column in ['CAMEO_DEUG_2015', 'CAMEO_INTL_2015']:
        df[column]=df[column].apply(str2float)

    #change the catergory columns into numbertraining
    for column in ["CAMEO_DEU_2015","D19_LETZTER_KAUF_BRANCHE","OST_WEST_KZ"]:
        df[column] = pd.Categorical(df[column])
        df[column] = df[column].cat.codes

    #extract the time,and keep the year
    df["EINGEFUEGT_AM"]=pd.to_datetime(df["EINGEFUEGT_AM"]).dt.year

    temp_col = ["ALTERSKATEGORIE_GROB","ANREDE_KZ","GEBAEUDETYP","GEOSCORE_KLS7","HAUSHALTSSTRUKTUR",
            'HH_EINKOMMEN_SCORE',"KBA05_BAUMAX","KBA05_GBZ","KKK","NATIONALITAET_KZ","PRAEGENDE_JUGENDJAHRE",
            "REGIOTYP","TITEL_KZ","WOHNDAUER_2008","WACHSTUMSGEBIET_NB","W_KEIT_KIND_HH"]
    for column in temp_col:
        try:
            df[column] = df[column].replace([0,-1],np.nan)
        except:
            pass
```

```
    temp_col= ['KBA05_ALTER1','KBA05_ALTER2','KBA05_ALTER3','KBA05_ALTER4','KBA05_ANHANG','KBA05_AUTOQUOT',
            'KBA05_CCM1','KBA05_CCM2','KBA05_CCM3','KBA05_CCM4','KBA05_DIESEL','KBA05_FRAU','KBA05_HERST1',
            'KBA05_HERST2','KBA05_HERST3','KBA05_HERST4','KBA05_HERST5','KBA05_HERSTTEMP','KBA05_KRSAQUOT',
            'KBA05_KRSHERST1','KBA05_KRSHERST2','KBA05_KRSHERST3','KBA05_KRSKLEIN','KBA05_KRSOBER','KBA05_KRSVAN',
            'KBA05_KRSZUL','KBA05_KW1','KBA05_KW2','KBA05_KW3','KBA05_MAXAH','KBA05_MAXBJ','KBA05_MAXHERST',
            'KBA05_MAXSEG','KBA05_MAXVORB','KBA05_MOD1','KBA05_MOD2','KBA05_MOD3','KBA05_MOD4','KBA05_MOD8',
            'KBA05_MODTEMP','KBA05_MOTOR','KBA05_MOTRAD','KBA05_SEG1','KBA05_SEG10','KBA05_SEG2','KBA05_SEG3',
            'KBA05_SEG4','KBA05_SEG5','KBA05_SEG6','KBA05_SEG7','KBA05_SEG8','KBA05_SEG9','KBA05_VORB0',
            'KBA05_VORB1','KBA05_VORB2','KBA05_ZUL1','KBA05_ZUL2','KBA05_ZUL3','KBA05_ZUL4','RELAT_AB',
            'SEMIO_DOM',
            'SEMIO_ERL','SEMIO_FAM','SEMIO_KAEM','SEMIO_KRIT','SEMIO_KULT','SEMIO_LUST','SEMIO_MAT',
            'SEMIO_PFLICHT',
            'SEMIO_RAT','SEMIO_REL','SEMIO_SOZ','SEMIO_TRADV','SEMIO_VERT','ZABEOTYP']
    for column in temp_col:
        try:
            df[column] = df[column].replace([9,-1],np.nan)
        except:
            pass

    for column in df.columns.values:
        try:
            df[column] = df[column].replace([-1],np.nan)
        except:
            pass

    df['GEBURTSJAHR'] = df["GEBURTSJAHR"].replace([0],np.nan)

    #get rid of the rows which have the age is 0 which corresponds to NaN
    df = df.drop(['Unnamed: 0'],axis = 1)
    if clustering:
      corr_matrix = df.corr().abs()
      limit = corr_matrix.where(np.triu(np.ones(corr_matrix.shape),k=1).astype(np.bool))
      drop_col = [column for column in limit.columns if any(limit[column] > 0.7)]
      df = df.drop(drop_col,axis =1)

    df = df.set_index('LNR')

    df_columns = list(df.columns.values)

    imputer = SimpleImputer(missing_values = np.nan , strategy = 'most_frequent')
    df = imputer.fit_transform(df)
    df = pd.DataFrame(df, columns = df_columns)
    print(df.shape)

    return df
```

I have done the following steps in pre-processing and cleaning

- Took the help of excel sheet to exactly know how unknown data is represented
- Filled the unknown data with NaNs
- Replace birth year with NaN, if it is zero.
- Dropped the columns, if most of the values are NaNs(only for unsupervised learning)
- Replaced NaN with most frequent value in that particular column

- Performed feature scaling
- If the cleaning is done for clustering, Remove the columns if most of the values are NaNs

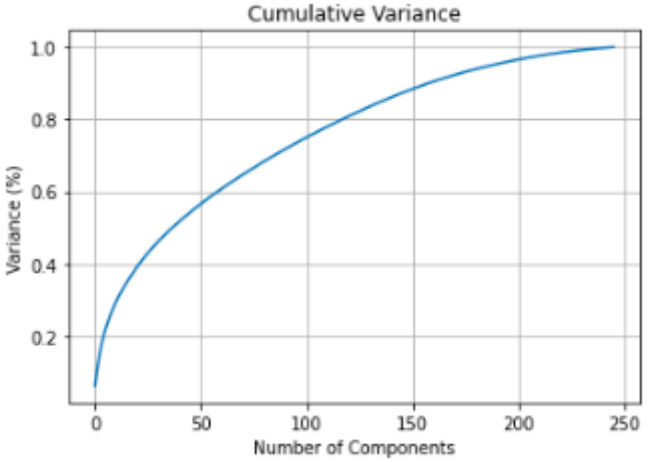
Customer Segmentation Report

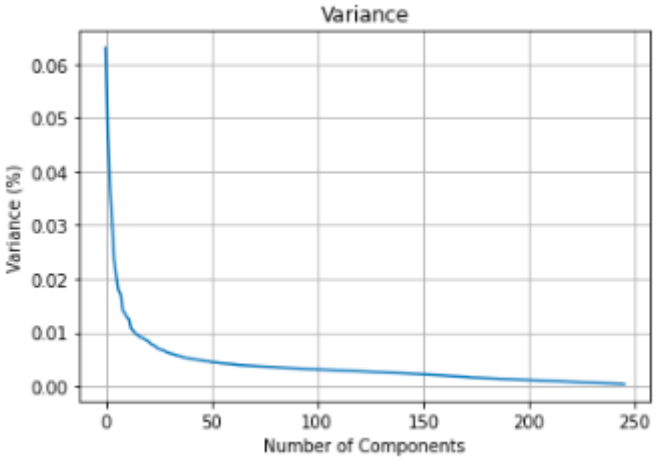
Key insights:

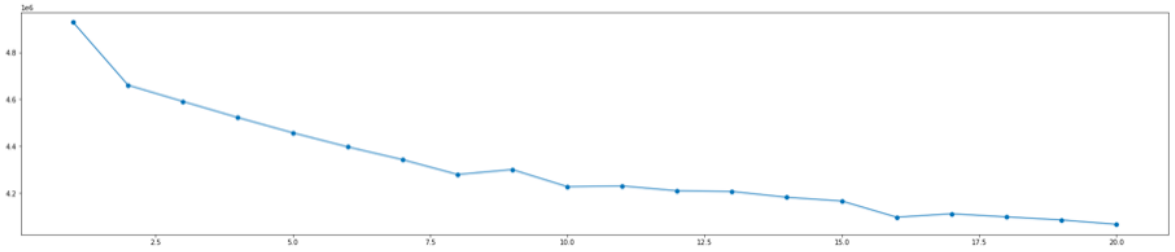
- There are clusters in the National Health and Nutrition Exam Survey (combined diet, medical, and exam datasets, 2013- 2014) which are only visible via dimensionality reduction.
- PCA in conjunction with k-means is a powerful method for visualizing high dimensional data.

Three steps done to reach this conclusion

- Step 1: Reduce Dimensionality
- Step 2: Find the Clusters
- Step 3: Visualize and Interpret the Clusters

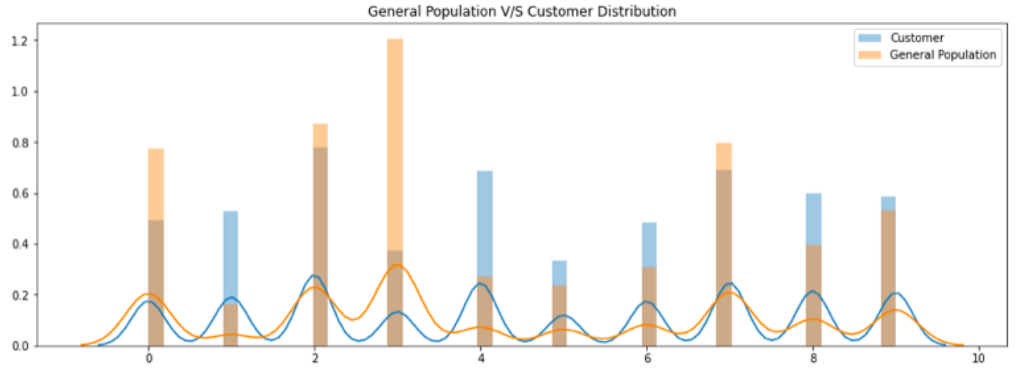






Graph with different centeroids for choosing best K-value in elbow method. For finding the best K value to apply on K means algorithm

Final Conclusion in K-Means clustering



The above graphs shows that K-Means clustering with K = 10 helps to find customer from general public, it has very small difference, compared to others

Conclusion on Unsupervised Learning.

The PCA gave me 200 components with a K Value of 10 gave me the best result.

Supervised Learning

Data preprocessing and cleaning is done. Since I have already developed a function for it.

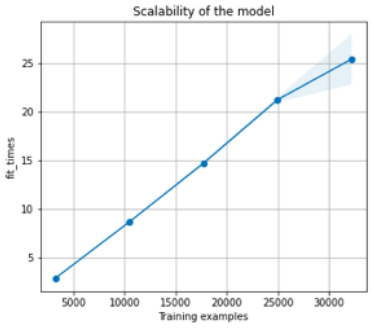
4 Different algorithms used to study about the supervised machine learning algorithms

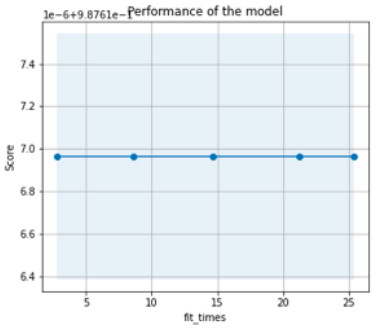
1. XGBoost,
2. Random Forest,
3. AdaBoost and
4. Decision Tree

plot_learning_curve is taken from sklearn's official site. And plotted to know how the different algorithms work on the dataset.

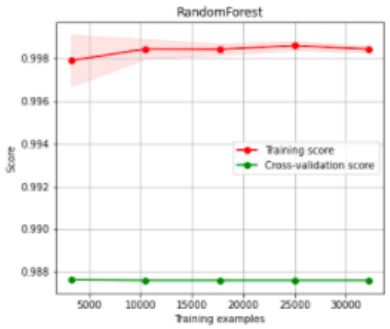
Learning Curve of different algorithms look like

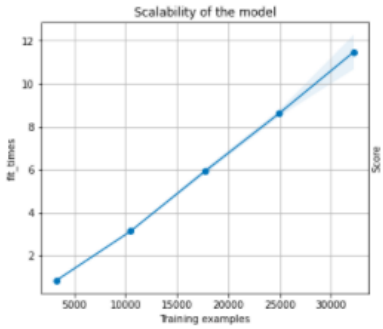
XGBoost

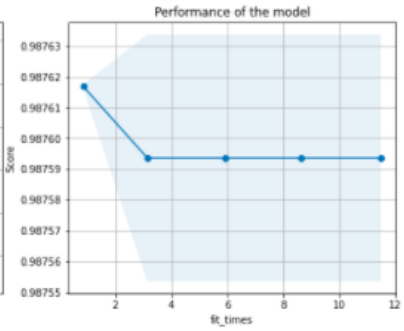




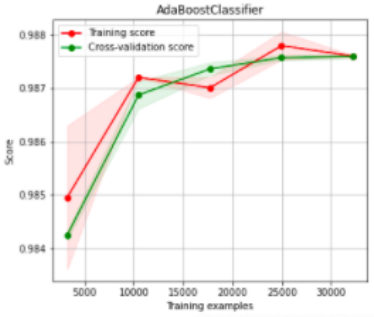
Random Forest,

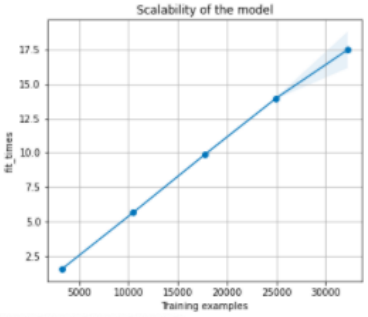


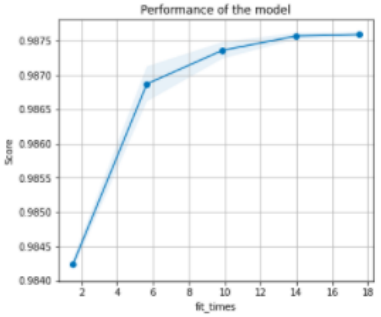




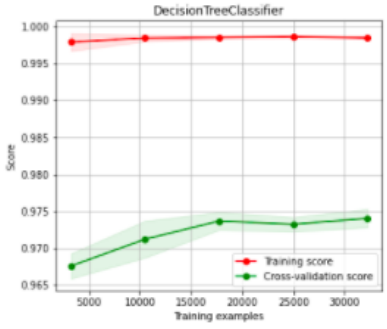
AdaBoost

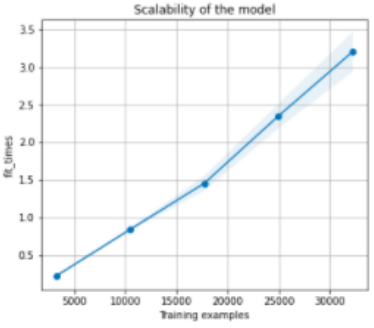


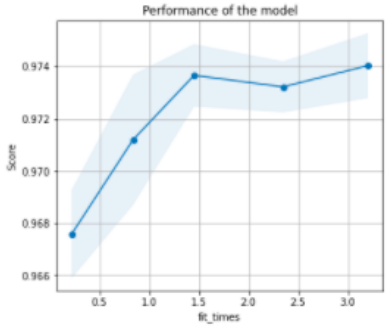




Decision Tree







And I found that XGBoost is performing well.

Hyperparameter Tuning

After hyperparameter tuning cross validation score came out to be 0.9876169638016635

Kaggle competition

The model was run on a provided dataset and then submitted on Kaggle and my model performed with a score 0.80596. And I grabbed 24th place in leadership board as on 20-09-2020