21/04/2017

Microsoft Professional Program for Data Science

Capstone Project

The Adventure Works Cycles Company

By: Sergio Díaz

The Adventure Works Cycles Company

Capstone Project

Executive Summary

This report presents the analysis performed on the data provided by the Adventure Works Cycles Co. (AWC Co.). The analysis is based on more than 18,000 observations of customer and sales information, each containing specific demographic characteristics of the customers and their associated buying track records. After exploring the data by calculating summary and descriptive statistics, and by creating visualizations, several relationships between customer features and their purchasing behavior were identified. Then, a predictive model to classify customers into "bike-purchaser" and "no-bike-purchaser" was created together with a regression model to predict the customer's average-monthly-expenditure for new customers based on their demographic characteristics.

After performing the data analysis, the following conclusions were reached:

- There are more bike buyers than not-buyers, with an overall 55% vs 45% proportion, respectively.
- Customers falling into Occupation category "Manual", Education level of "Partial High School", age younger than 30 or yearly income lower than 50K does not typically buy bikes.
- Married or male customers are more likely to buy bikes than single or female individuals.
- Customers that have children at home are more prone towards buying a bike than those who do not.
- Customer's yearly income is highly correlated with degree of education.
- About 50% of all customers have a yearly income in between 50-90K.
- About 40% of the customers reside in USA, followed by 30% in Europe and 20% in Australia.
- In the USA and Canada customers are primarily concentrated in the east coast in the states of California, Washington Oregon and British Columbia, respectively.
- Customers are not evenly distributed throughout the territory but are concentrated mainly at certain states/provinces, primarily within major urban locations.
- Customer expenditure is highly correlated with Income and Occupation. Male individuals in the 30-50 age range tend to spend more than the rest of the population included in the database.

Table of Contents

Introduction

The Adventure Works Cycles Company (AWC Co.) collected a large volume of data about their existing customers, including personal and contact details as well as purchases they have made in the past. The company requirement is to determine apparent relationships between demographic features known about the customers in order to predict the likelihood of future customers purchasing a bike. Additionally, it is also desirable determining in advance customer's average monthly expenditure based on customer profiling. The information gained can then be used to define and activate specific commercial strategies to increase sales or to optimize current resources while maintaining or increasing profits.

In this report the datasets described below were examined using data analytics tools including: codes written in Python language, MS Excel and Power BI software, and the MS Azure Machine Learning platform. The main objectives of this work were therefore to create:

- A classification model that predicts whether or not a new customer will buy a bike.
- A regression model that predicts the average monthly expenditure of new customers.

In a first delivery, two CVS (comma separated values) format files were supplied by AWC Co. staff as input for this project. One file (AWCustomers) contained customer personal information while a second file (AWSales) included sales transactions only. Relationship between both files was established using customer's ID as common key. The AWCustomers file contained 1861 records comprising 24 characteristics (also used here as features or attributes) per customer, whereas AWSales file included 18355 rows or transactions with only three features per record. The original datasets are described in detail in Appendix A.

In a second delivery, the AWC Co. staff provided two additional CVS format files. These files contained a total of 75 customer records with the same type of personal attributes as in the first input. However, in this occasion the corresponding AWSales file having commercial transactions was not supplied. By so doing, AWC Co. was able to evaluate and quantify the predictive capacity of the machine learning models developed in this work.

The raw customer's database had a number of missing values and duplicate records, as typically found in real world databases. Initial data cleaning and preparation consisted in the removal of duplicate records; the preliminary identification of attributes significance, dependencies and redundancies; as well as the counting unique values per attributes. The information was then explored calculating summary and descriptive statistics of the most relevant features and by creating visualizations to discover apparent relationships in the data.

Preliminary Data Exploration

Who are our customers?

A preliminary data analysis was performed to study the overall characteristics of the customers and to better understand the information contained in the company´s database. The AWC Co. has currently 18355 registered customers with a wide variety of social, economic, and demographic characteristics, some of which are graphically depicted in Figure 1. By exploring the plots shown in the figure, the following key statements can be drawn:

- Customer´s gender distribution is almost balanced in a 50/50% fashion.
- There are slightly more married (54%) than single customers.
- About 60% of the customers have 40 year old or less.
- About 80% of the customers own at least one car.
- About 50% of the population does not have children.
- The top occupation category among customers is that of “Skilled Manual” representing a share of 33%. On the opposite end is the “Professional” category with almost 9% of the total population.

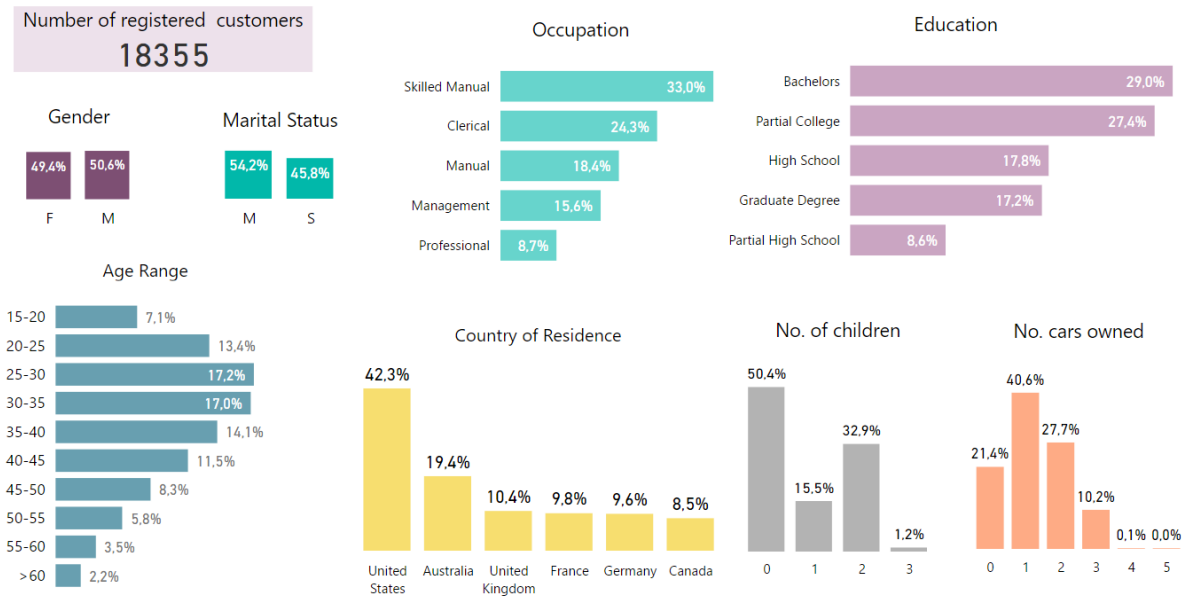

Figure 1. Customer distribution per social characteristics.

Figure 2 shows basic apparent correlations between the median yearly income and some of the principal customer´s features. The following key statements can be drawn from the plots included in this figure:

- There is a clear correlation between yearly income and the different categories within the attribute occupation. Those customers in the category “professional” are at the top of the income scale.
- Customer’s yearly income is also highly correlated with degree of education.
- Customers that own a house tend to earn more than non-owners.
- Number of cars owned by customers is correlated with their yearly income.
- About 50% of all customers have a yearly income in between 50-90K.

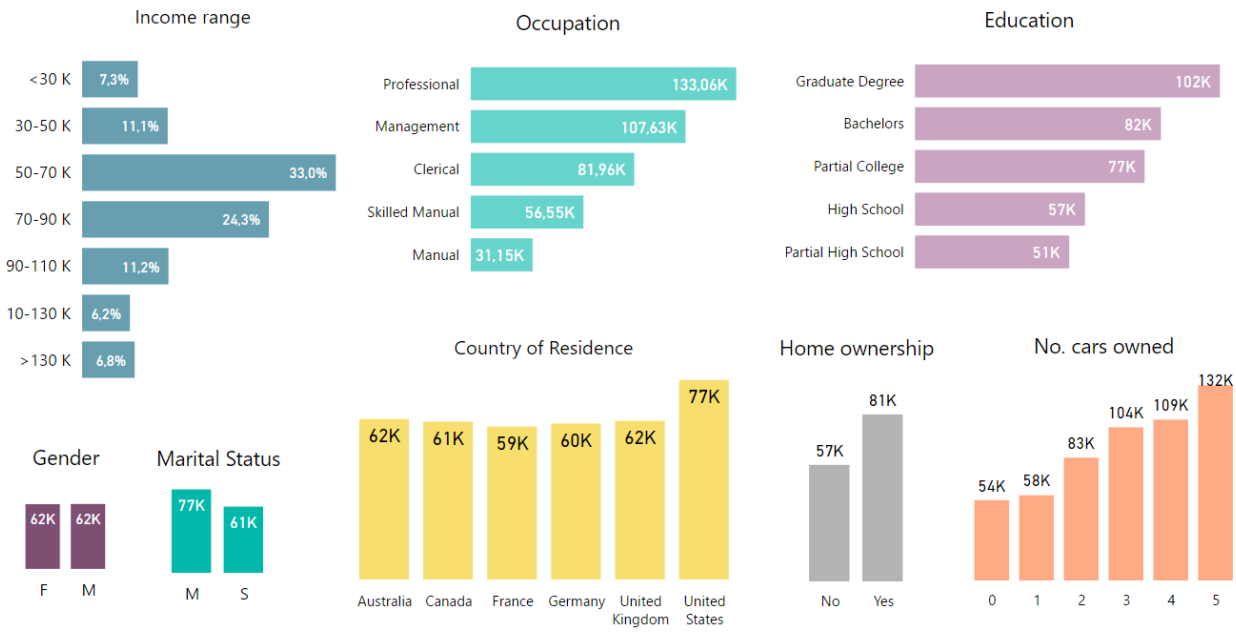

Figure 2. Median yearly income with respect to customers' social characteristics.

Where do our customers live?

As shown in Figure 1, about 40% of the AWC Co. customers reside in USA, followed by Australia with almost 20%, Europe with 30% and Canada with slightly over 8%. In the USA customers are concentrated in the east coast in the states of California, Washington and Oregon.

Common to all countries, clients are not evenly distributed throughout the territory but are concentrated mainly at certain states/provinces and primarily within major urban locations. Figure 3 gives a better view of customer´s geographic location, where circles size and color shown in the maps are proportional to number of customers.

Figure 3(a). Customer's country of residence (circle size/color proportional to customers count).

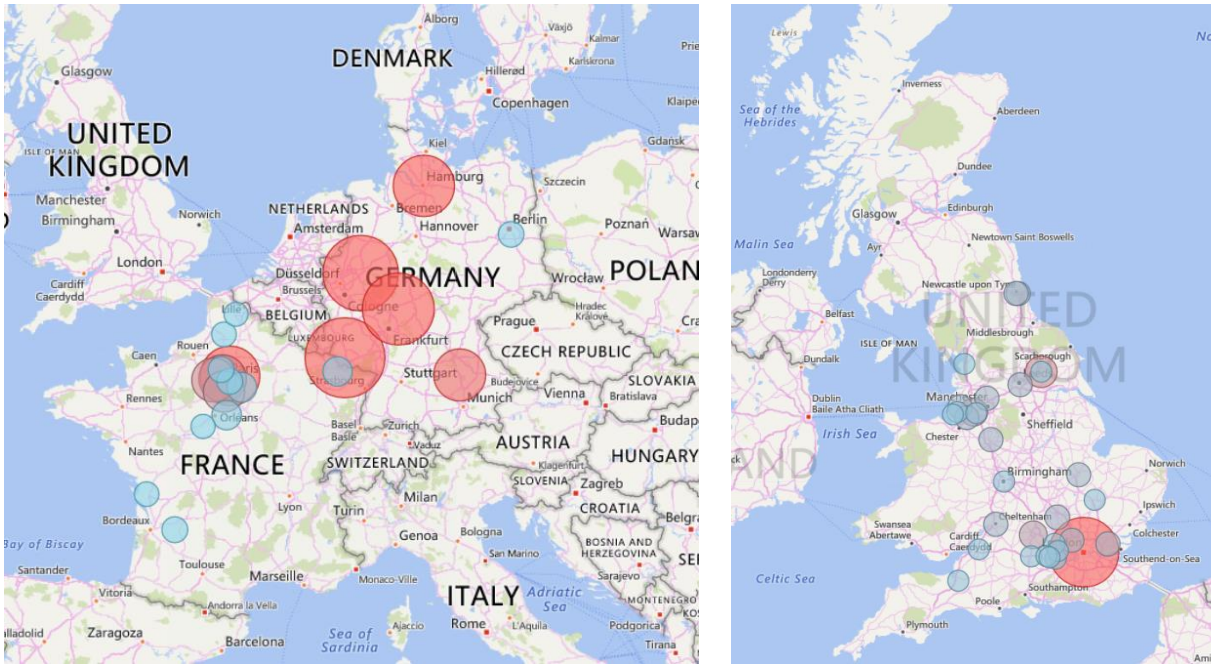

Figure 3(b). Customer's country of residence (circle size/color proportional to customers count).

Data Analysis

Who buys our bikes?

Since the objective of the project was to predict the purchasing behavior of the customers, it is then important to understand the apparent correlation between customer's main descriptors and their purchasing track records. Figure 4(a-e) are devoted to comparing the proportion of bike buyers (green) vs. non-buyers (red) for each feature contained in the company´s database. The rationale behind this effort was to identify the most relevant characteristics to be employed for classification.

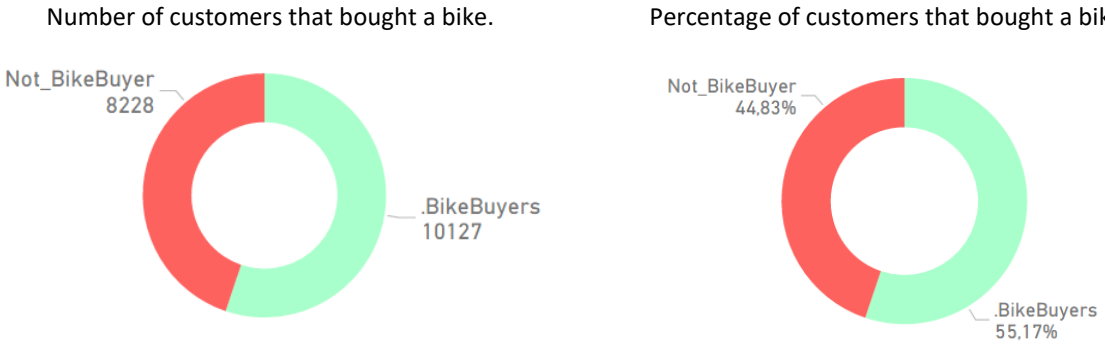

Figure 4(a). Customer proportion of bike buyers (green) vs. not-buyers (red).

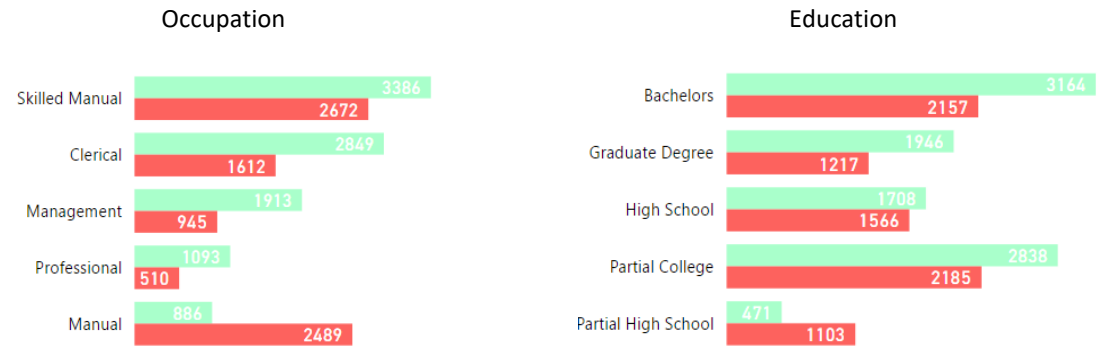

Figure 4(b). Bike buyers vs. not-buyers as a function of occupation and degree of education.

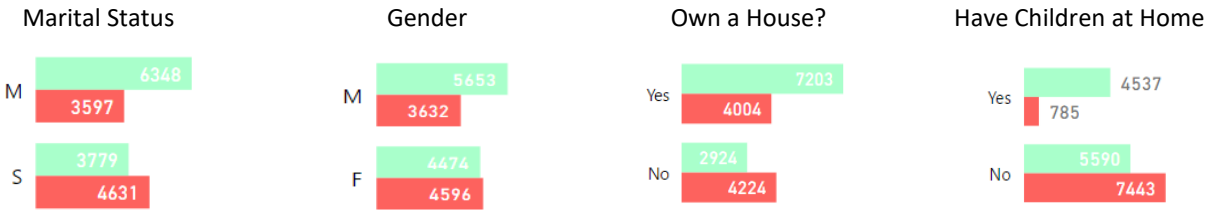

Figure 4(c). Bike buyers vs. not-buyers per marital status, gender, house ownership and children.

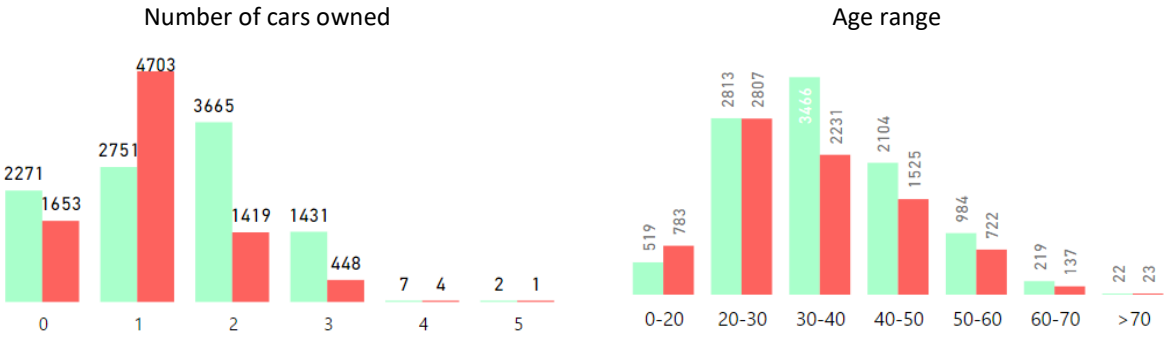

Figure 4(d). Number of bike buyers vs. not-buyers per number of cars owned and customer's age.

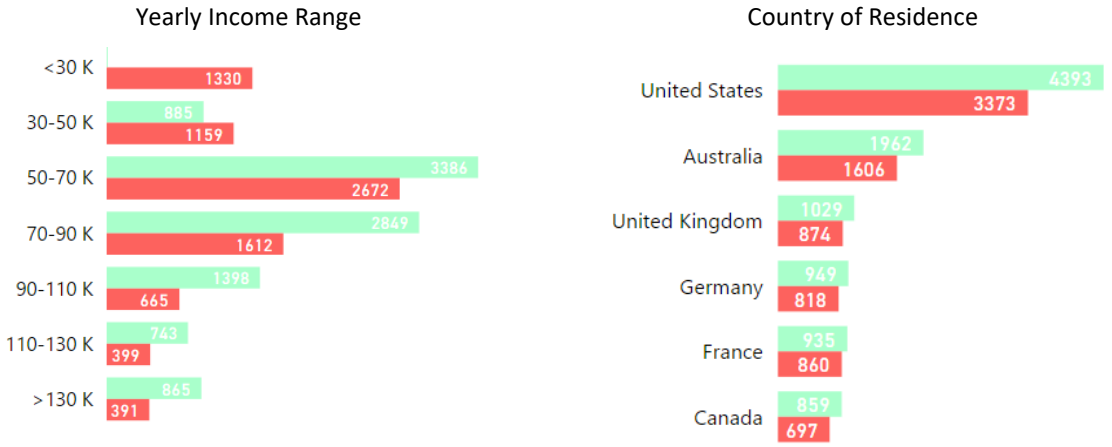

Figure 4(e). Number of bike buyers vs. not-buyers per yearly income and country of residence.

By exploring Figure 4(a-e) the following observations can be drawn:

- There are more bike buyers than not-buyers, with an overall 55% vs 45% proportion, respectively.
- There are more bike buyers than non-buyers for most Occupation categories, except for those customers in the "Manual" category, which show a significant tendency for being not-bike buyers. Therefore, this feature is likely to be included in any purchasing prediction model.
- There are more bike buyers than non-buyers for most Education categories, except for those customers in the "Partial High School" category, which show a significant tendency for being not bike buyers. Therefore, this feature is likely to be included in any purchasing prediction model.
- Married customers are more likely to buy bikes than single customers.
- Male customers are more likely to buy a bike than female customers do.
- Customers that own a house are more likely to buy a bike than customers that do not have a property.
- Customers that have children at home are more prone towards buying a bike than those who do not have children at home.
- For unknown reasons, customers that have one car are less likely to purchase a bike when compared with the rest of categories, even than those who do not have a car at all.
- Customers younger than 30 are less likely to purchase a bike than the rest of the population in the database.
- Customers with a yearly income lower than 50K are less likely to buy bikes than the rest of the population.

These are just some of the "straight-forward" remarks that can be stated by inspecting the figure. However, since customers' characteristics are not independent of each other, the above listed statements were taken only used as guidance towards gaining insight to better predict the purchasing behavior.

Where do our bike-buyers live?

Table 1(a) lists the number of total customers and bike-buyers for each country. As previously shown in Figure 1, about 42% of all AWC Co. customers live in the USA for a total count of 7766 individuals. From those living in USA, almost 4400 are bike buyers representing 57% of all USA customers. This percentage is only slightly above the global mean of 55% (Bike Buyers / Total Customers). This small difference is also displayed by the other four countries, meaning that the ratio between bike buyers to non-buyers remain almost constant among the five nations. This fact preludes a low significance of the "Country" feature in the building of a purchasing predicting model.

Table 1(a). Number of total customers, number of bike-buyerr, and bike-buyers % per country.

Country	Customers	BikeBuyers	BikeBuyers%
United States	7766	4393	57 %
Australia	3568	1962	55 %
United Kingdom	1903	1029	54 %
Germany	1767	949	54 %
France	1795	935	52 %
Canada	1556	859	55 %
Total	**18355**	**10127**	**55 %**

Table 1(b). Top5 state/provinces per country in the number of bike-buyer-customers.

USA

StateProvinceName	BikeBuyers
California	2497
Washington	1281
Oregon	592
Illinois	4
Ohio	4

Australia

StateProvinceName	BikeBuyers
New South Wales	838
Victoria	492
Queensland	425
South Australia	147
Tasmania	60

United Kingdom

StateProvinceName	BikeBuyers
England	1029

Germany

StateProvinceName	BikeBuyers
Saarland	240
Nordrhein-Westfalen	221
Hessen	200
Hamburg	158
Bayern	111

France

StateProvinceName	BikeBuyers
Seine (Paris)	203
Seine Saint Denis	146
Nord	143
Hauts de Seine	104
Essonne	86

Canada

StateProvinceName	BikeBuyers
British Columbia	853
Alberta	6
Ontario	0

Table 1(b) lists the top five state/provinces per country with respect to number of bike buyers. These numbers indicate once more that bike buyers and customers in general reside at specific locations.

How much do customers spend in our store?

Figure 5 shows a scatter plot of monthly average expenditure vs. yearly income for all customers in the database. This figure clearly reveals that there exists a very specific pattern of the customers 'YearlyIncome' feature, where income is arranged in five well-defined, non-overlapping bands or groups. Table 2 lists the maximum, minimum and range for each one of these groups. Further analysis of the data revealed that there is a one-to-one relationship between the five categories comprising the "Occupation" feature and the five income groups, where the "Professional" Occupation category receives the highest income and at the lower end the "Manual" Occupation category is linked with the lowest income group. There are no cross-overs between Occupation categories and income groups.

Figure 5. Mean monthly expenditure vs yearly income for all customers (shown as circles).

Table 2 . Occupation and yearly-income-group maximum and minimum values together with range.

Occupation	IncomeGroup	.IncomeMax	.IncomeMin	.IncomeRange
Professional	5	139115	127166	11949
Management	4	113674	101730	11944
Clerical	3	88226	76294	11932
Skilled Manual	2	62806	50869	11937
Manual	1	37374	25435	11939

A closer look to table 2 reveals two specific characteristics of the income groups. Firstly, the fact that separation between income groups is fixed to multiples of $25,400. Secondly, that income group range in all cases can be rounded off to $12,000. These two characteristics were used to "collapse" the information contained in figure 5 onto figure 6, where a sort of normalized or "Unified" Yearly Income was used in a scatter plot together with customer's average monthly expenditure. UnifiedYearlyIncome is thus related to YearlyIncome in the following fashion:

UnifiedYearlyIncome = YearlyIncome - (IncomeGroup *25400)

This data processing is aimed at reducing data dimensionality by eliminating redundancies between features.

Figure 6. Mean monthly expenditure vs Unified yearly income for all customers (represented as circles).

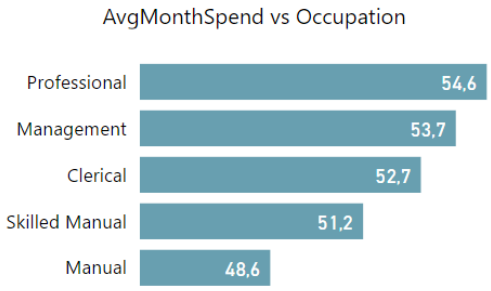

Figure 7. Mean monthly expenditure ($) vs Occupation.

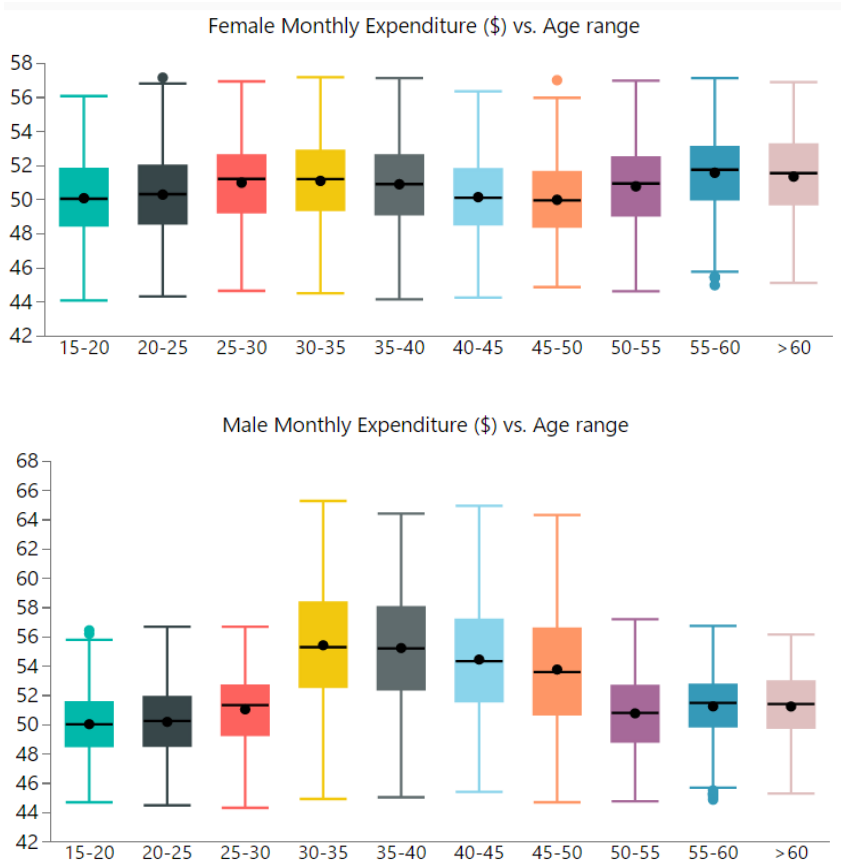

Figure 8. Female monthly (top) and Male (bottom) expenditure ($) vs Age range.

Figure 7 shows a clear correlation between 'Occupation' and monthly expenditure. Additionally, by inspecting Figure 8 it can be seen that monthly expenditure by female customers remains more or less constant at all age ranges whereas a purchasing spike can be detected on males in the 30-50 age range.

Based on the data exploration performed so far, customer features were classified as having high, medium or low relevance as predictors of customers' purchasing behavior, as listed in Table 3. Features that showed changes in any given purchasing pattern among the different categories comprising a feature were classified of high relevance. Furthermore, only those features classified as relevant were initially included in the development of a machine learning predictive model.

Table 3. Customer's database features classification by predicting relevance.

Relevance	Feature Name
High	Age (derived from BirthDate), Education, Occupation, Gender, MaritalStatus, HomeOwnerFlag, NumberCarsOwned, NumberChildrenAtHome, TotalChildren, UnifiedYearlyIncome (derived from Yearly Income)
Medium	City, StateProvince, CountryRegion
Low	Title, CustomerID, FirstName, MiddleName, LastName, Suffix, AddressLine1, AddressLine2, PostalCode, PhoneNumber, LastUpdated

Building a Machine Learning Predictive Model

Customers Classification into Bike Buyer/Non-Buyer

Since the first objective of this work was to create a numerical model to classify customers into bike buyers or non-buyers, then a two-class supervised learning algorithm was selected to model customer behavior. Using the Microsoft Azure Machine Learning Algorithm Cheat Sheet [1] as guideline, a DecisionTreeClassifier together with a RandomForestClassifier algorithms were selected for the task. The Python code **BikeBuyerClassifier.py** listed in Appendix B was thus written and implemented.

As a supporting tool in the classification process, the Python code **BikeBuyerSlicer.py** was also written. Its basic operation is to use each row in the AWTest-Classification file as filter/slicer of the dataset contained in the AWCustomers file. The filtering is performed 50 times; the number of customers in AWTest-Classification. On each iteration, the number of proportion of buyers to non-buyers is calculated summing the 'BikeBuyer' column with row values equal 1.

The **BikeBuyerSlicer.py** and the **CustomerExpenditureSlicer.py** (discussed below) codes are better explained visually with the aid of Figure 9, which is the result of applying a given set of filters to Figure 6. What is left in the Expenditure vs Income plot shown are those individuals that comply with the defined filter context. In this example, filters have been set to the features values of CustomerID 11908, classified thus as BikeBuyer with an AvgMonthSpend of about $58.80.

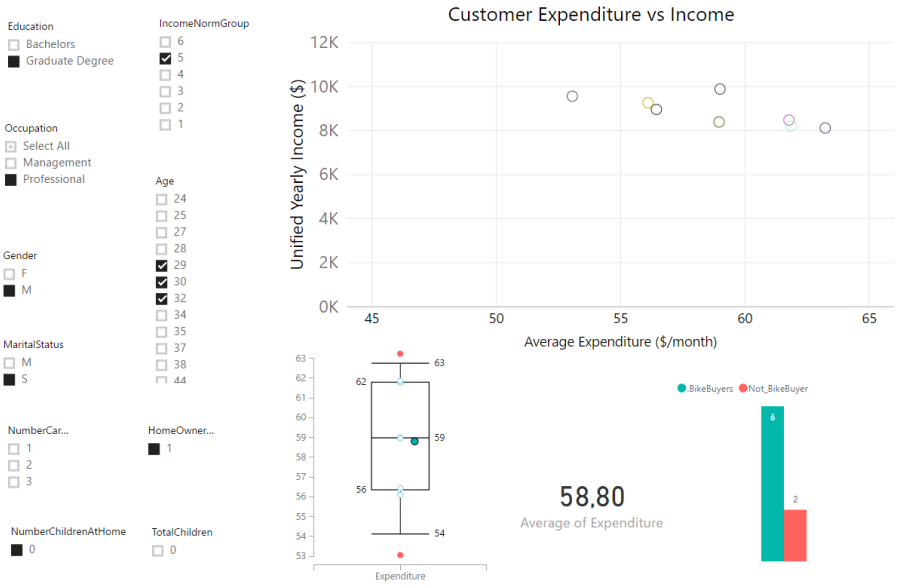

Figure 9. CustomerID 11908 estimated Avg monthly expenditure and classification as BikeBuyer based on data from individuals with similar characteristics.

Customer Monthly Expenditure Estimation

For the regression model to predict the average monthly expenditure of new customers, a decision tree regression algorithm was selected following the guidelines found at the Scikit Learn website [2]. The Python code **CustomerExpenditureRegression.py** listed in Appendix B was thus written and implemented.

Also as previously done with classification, an alternative procedure (**CustomerExpenditureSlicer.py)** to the machine learning algorithm was developed. It uses each row of the AWTest-regression file as filter/slicer of data in AWCustomers file. This filtering is performed 25 times (the number of customers in AWTest-regression). On each iteration, the mean of 'AvgMonthSpend' is calculated out the remaining rows after the filtering procedure.

Testing the Machine Learning Predictive Model

Figure 10 and Figure 11 show an example of the output produced by the Machine Learning classification and regression codes listed in Appendix B, respectively.

```
RandomForest Score:79.992
Decision Tree Score:80.373
Features Rank:                                rank
NumberChildrenAtHome_0          0.264842
NumberCarsOwned_1               0.202858
Occupation_Manual               0.160892
IncomeNorm                      0.108975
MaritalStatus_S                 0.061909
Education_Partial High School   0.056541
Gender_M                        0.046291
Age                             0.033456
NumberCarsOwned_2               0.030728
Gender_F                        0.020385
MaritalStatus_M                 0.012776
Occupation_Skilled Manual       0.000347
Occupation_Management           0.000000
NumberCarsOwned_3               0.000000
NumberChildrenAtHome_2          0.000000
NumberChildrenAtHome_1          0.000000
Education_Bachelors             0.000000
NumberCarsOwned_5               0.000000
NumberCarsOwned_4               0.000000
Education_Graduate Degree       0.000000
Occupation_Clerical             0.000000
NumberCarsOwned_0               0.000000
Education_High School           0.000000
Education_Partial College       0.000000
Occupation_Professional         0.000000
NumberChildrenAtHome_3          0.000000
Random Forest Prediction:[1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1
 1 1 1 1 1 1 0 1 1 1 0 1 1]
Decision Tree Prediction:[1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1
 0 1 1 1 1 1 0 1 1 1 0 1 1]
```

Figure 10. Preliminary test results for classification using **BikeBuyerClassifier.py**

```
Decision Tree Score:65.448
Features Rank:                                rank
Age                             0.308195
Occupation_Manual               0.280719
Occupation_Skilled Manual       0.122171
Gender_F                        0.080301
Gender_M                        0.068211
NumberChildrenAtHome_0          0.051384
Occupation_Clerical             0.028239
MaritalStatus_M                 0.021813
MaritalStatus_S                 0.016184
IncomeNorm                      0.012999
NumberChildrenAtHome_1          0.003447
Occupation_Management           0.003363
Occupation_Professional         0.002077
Education_Bachelors             0.000296
Education_Partial College       0.000250
Education_Graduate Degree       0.000229
Education_High School           0.000122
Education_Partial High School   0.000000
NumberChildrenAtHome_2          0.000000
NumberChildrenAtHome_3          0.000000
Decision Tree Prediction:
[ 49.05053333  49.05053333  49.05053333  50.31865854  48.36216216
  48.81103448  52.95290323  52.77275132  55.02865079  57.11068273
  52.52364706  51.95864706  52.37015873  54.34849206  51.95864706
  54.34849206  55.52068093  55.02865079  52.94091603  54.34849206
  53.31433692  50.82772321  52.09997455  55.06473118  52.94091603]
```

Figure 11. Preliminary test results for regression using **CustomerExpenditureRegression.py**.

The initial versions of both the classification and regression algorithms included all of the features identified as of high relevance back in Table 3. Following the initial execution of both ML algorithms, some features were removed according to the feature_importance output. Thus, 'HomeOwnerFlag' and 'TotalChildren' were not used for both classification and regression tasks, and 'NumberCarsOwned' was also not considered for regression.

Table 4 and Table 5 list the progress made on predicting customers purchasing behavior for those customer ID´s listed in AWT-classification and AWT-regression files. Results for the prediction of Average Monthly Spend improved progressively on each iteration. However, that was not the case for the classification of bike buyers, where score diminished from iteration 3rd to 4th due to a mistake in the Python Code identified afterwards.

Due to the early stage of development and low scoring of the both ML algorithms, the classification and regression results obtained via machine learning were combined with the output produced by the supporting codes using the slicing/filtering method described earlier.

Table 4. Bike Buyer classification results

	Iteration Number							
CustomerID	1	2	3	4	6	7	8	9
11219	1	1	1	1	1			
11241	1	0	0	0	0			
11352	1	0	0	1	0			
11904	1	1	1	0	1			
11908	1	1	1	1	1			
12113	1	1	1	1	1			
12221	1	0	0	0	0			
12286	1	1	1	1	1			
12292	1	0	0	0	0			
12305	1	1	1	1	1			
12807	0	0	0	0	0			
12820	0	0	0	0	0			
12892	1	1	1	1	1			
13310	1	1	1	1	1			
13349	1	1	1	1	1			
13482	1	1	1	1	1			
13640	1	1	1	0	1			
13838	1	1	1	0	1			
14320	1	1	1	1	1			
14327	1	0	0	0	0			
14380	1	1	1	1	1			
14477	1	1	0	0	0			
14564	1	0	1	1	0			
14652	1	1	1	1	1			
14698	1	1	0	0	0			
14778	0	0	0	0	0			
15085	1	1	1	1	1			
15177	1	1	1	1	1			
15388	1	1	1	1	1			
15464	1	1	1	1	1			
15573	1	1	0	1	1			
15626	1	1	1	1	1			
15727	1	1	1	1	1			
15738	1	1	1	1	1			
15866	0	0	0	0	0			
15875	1	1	1	1	1			
16184	1	1	1	1	1			
16394	0	0	0	0	0			
16575	1	0	1	0	0			
17536	1	1	1	0	1			
17605	1	1	1	1	1			
17723	1	0	0	0	0			
17726	1	0	1	0	0			
18171	0	0	0	0	0			
18345	1	0	0	1	0			
18382	1	0	1	0	0			
18842	1	1	1	0	1			
18956	0	0	0	0	0			
19019	1	1	1	1	1			
19315	1	1	1	0	1			
Score	36	37	40	35	39			

Table 5. Average monthly spend regression results

	Iteration Number								
CustomerID	1	2	3	4	5	6	7	8	9
11908	55.61	54.00	58.00						
12286	54.26	54.26	54.26						
12892	52.28	52.28	52.28						
14698	51.27	51.27	51.27						
15085	54.30	54.30	54.30						
15727	53.40	53.40	53.40						
17418	52.30	52.30	52.30						
17723	51.30	51.30	51.30						
18644	53.15	53.15	53.15						
19218	56.00	54.00	57.00						
20671	53.00	53.00	53.00						
20697	52.47	52.47	52.47						
20734	51.40	51.40	51.40						
20812	51.50	50.50	49.00						
23244	54.30	54.30	54.30						
23454	52.60	54.00	55.00						
24420	56.00	57.00	57.00						
25193	52.00	52.00	52.00						
25793	51.50	50.40	52.50						
26410	55.00	54.50	53.00						
26694	53.00	57.00	57.00						
28096	51.20	51.20	51.20						
28676	53.00	57.00	55.00						
29015	52.70	52.70	52.70						
29377	53.60	53.60	53.60						
Score	13	19	22						

Conclusions

In this report a dataset containing customer demographic information and purchasing transactions was analyzed. Codes written in Python Language and the use of data analysis software such as MS Excel, Power BI and the Microsoft Azure Machine Learning Platform were used for data exploration and visualization and to understand relationships between the data columns. Redundant collinear features were removed and key features identified that were then used for predicting customer purchasing behavior in two supervised machine learning algorithms.

The numerical models obtained here are still at an early stage of development with a prediction score of 80% for classification and 65% for regression. Test results produced by the ML models were compared and combined with those obtained with a conventional but effective procedure of filtering and grouping the principal dataset followed by basic statistics performed on the remaining data slice.

Future Work

As of the date of publication of this report, a preliminary version of the Machine Learning Models has been produced with only fair prediction capabilities. Further fine-tuning of the models is needed and schedule in the upcoming weeks. There is plenty of room for improvement since areas such as: cross validation, identification and removal of outliers, dimensionality reduction, algorithm selection and parameters adjustment have not been thoroughly exploited due to time constraints.

References

[1] https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice. Accessed 10 Apr. 2017

[2] http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html. Accessed 10 Apr. 2017

Appendix A: Datasets

Table A1. Files supplied by AWC Co. as input for the project.

Filename:	AWCustomers.csv	AWTest-Classification.csv	AWTest-Regression. csv	AWSales.csv
Number of instances:	18361	50	25	18355
Number of attributes:	24	24	24	3
Missing values:	Yes	Yes	Yes	No
Scope:	Customer demographic data			Commercial transactions

Table A2. Attributes included in files AWCustomers, AWTest-classification and AWTest-Regression:

Attribute Name	Data Type	Description
Title	string	The customer's formal title (Mr, Mrs, Ms, Miss Dr, etc.)
CustomerID	integer	A unique customer identifier.
FirstName	string	The customer's first name.
MiddleName	string	The customer's middle name.
LastName	string	The customer's last name
Suffix	string	A suffix for the customer name (Jr, Sr, etc.)
AddressLine1	string	The first line of the customer's home address.
AddressLine2	string	The second line of the customer's home address.
City	string	The city where the customer lives.
StateProvince	string	The state or province where the customer lives.
CountryRegion	string	The country or region where the customer lives.
PostalCode	string	The postal code for the customer's address.
PhoneNumber	string	The customer's telephone number.
BirthDate	date	The customer's date of birth in the format YYYY-MM-DD. Used to derive and replaced by the feature "Age"
Education	string	The maximum level of education achieved by the customer: • Partial High School • High School • Partial College • Bachelors • Graduate Degree
Occupation	string	The type of job in which the customer is employed: • Manual • Skilled Manual • Clerical • Management • Professional
Gender	string	The customer's gender (for example, M for male, F for female, etc.)
MaritalStatus	string	Whether the customer is married (M) or single (S).
HomeOwnerFlag	integer	A Boolean flag indicating whether the customer owns their own home (1) or not (0).
NumberCarsOwned	integer	The number of cars owned by the customer.
NumberChildrenAtHome	integer	The number of children the customer has who live at home.
TotalChildren	integer	The total number of children the customer has.
YearlyIncome	decimal	The annual income of the customer. Used to derive and replaced by the feature "Unified Yearly Income"
LastUpdated	date	The date when the customer record was last modified.

Table A3. Attributes included in file AWSales:

Attribute Name	Data Type	Description
CustomerID	integer	The unique identifier for the customer.
BikeBuyer	integer	A Boolean flag indicating whether a customer has previously purchased a bike (1) or not (0).
AvgMonthSpend	decimal	The amount of money the customer has spent with Adventure Works Cycles on average each month.

Appendix B: Codes

BikeBuyerClassifier.py

```
# Import Modules
import pandas as pd
import numpy as np
import os

# Features used in the classification model:
# 'CustomerID','Education','Occupation','Gender','MaritalStatus','NumberCarsOwned',
# 'NumberChildrenAtHome','TotalChildren','HomeOwnerFlag','Age','IncomeNorm',
# 'BikeBuyer', 'AvgMonthSpend'

# Read Customers Database
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersDatabase002.csv'
pathfile= os.path.join(path,filename)
df = pd.read_csv(pathfile, sep=',', index_col=0)

df.reset_index(inplace=True, drop=True)
dropcols = ['AvgMonthSpend','TotalChildren','HomeOwnerFlag']
df.drop(dropcols, axis=1, inplace=True)
data=df

# Read  dataset to Classify (AWTest-classification)
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersTest002.csv'
pathfile= os.path.join(path,filename)
df_class = pd.read_csv(pathfile, sep=',', index_col=0)
df_class = df_class.ix[SelectedCustomers]
df_class.reset_index(inplace=True,drop=True)
dropcols = ['TotalChildren','HomeOwnerFlag']
df_class.drop(dropcols, axis=1, inplace=True)
#df_class

# Remove labels from main dataframe
y_data = data['BikeBuyer'].copy()
data.drop(['BikeBuyer'], axis=1, inplace=True)

# Prepare X dataframes
dummies=['Education',
         'Occupation',
         'Gender',
         'MaritalStatus',
         'NumberCarsOwned',
         'NumberChildrenAtHome']
         #'HomeOwnerFlag', removed since not significant in feature importance
         #'TotalChildren'] removed since not significant in feature importance
x_data = pd.get_dummies(data, columns=dummies)
x_class = pd.get_dummies(df_class, columns=dummies)
# Padd with Zeros missing columns in x_class
for col in x_data.columns:
    if col not in x_class.columns:
        x_class[col]=0

# Split data into train and test
from sklearn.model_selection import train_test_split
(xtrain, xtest, ytrain, ytest) = train_test_split(x_data, y_data, test_size=0.4, random_state=0)

# RANDOMFOREST ==================
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=30,oob_score=True, random_state=0, max_depth=7)
model.fit(xtrain, ytrain)
oobscore = model.oob_score_
RFscore= model.score(xtest,ytest)
RFpredict=model.predict(x_class)

# DECISION TREE==========
from sklearn import tree
model = tree.DecisionTreeClassifier(max_depth=6, random_state=0,criterion='entropy',splitter='best')
model.fit(xtrain,ytrain)
featureimp=pd.DataFrame(model.feature_importances_,index=x_data.columns,columns=['rank'])
topfeatures= featureimp.sort_values('rank', ascending=False)
DTscore=model.score(xtest,ytest)
DTpredict=model.predict(x_class)

# RESULTS
print "RandomForest Score:{}".format(round(RFscore*100, 3))
print "Decision Tree Score:{}".format(round((DTscore*100),3))
print "Features Rank:{}".format(topfeatures)
print "Random Forest Prediction:{}".format(RFpredict)
print "Decision Tree Prediction:{}".format(DTpredict)
```

BikeBuyerSlicer.py

```
#
import pandas as pd
import numpy as np
import sklearn.metrics as metrics
import os

# read Customers Database (AWCustomers)
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersDatabase002.csv'
pathfile= os.path.join(path,filename)
db = pd.read_csv(pathfile, sep=',', index_col=0)
db.reset_index(inplace=True, drop=True)
dropcols = ['AvgMonthSpend']
db.drop(dropcols, axis=1, inplace=True)

# Read unseen dataset to Classify (AWTest-classification)
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersTest002.csv'
pathfile= os.path.join(path,filename)
dfclass = pd.read_csv(pathfile, sep=',', index_col=0)
dfclass.reset_index(inplace=True,drop=True)

# Features used in the code
# ['CustomerID','Education','Occupation','Gender','MaritalStatus','NumberCarsOwned','NumberChildrenAtHome',
# 'TotalChildren','HomeOwnerFlag','Age','IncomeNorm', 'BikeBuyer', 'AvgMonthSpend' ]

# Loop over all rows of the Classification Test files
for i in range(50):
    df=db
    headers = dfclass.columns
    for col in headers:
        value=dfclass.loc[i,col]
        if col=='Age':
            age_limit = 3 # number that determines range of age window
            age_lower = value - age_limit
            age_upper = value + age_limit
            mask= (df[col]>= age_lower) & (df[col]<= age_upper)
        elif col=='IncomeNorm':
            Income_limit = 1000 # number determining range of income window
            Income_lower = value - Income_limit
            Income_upper = value + Income_limit
            mask= (df[col]>= Income_lower) & (df[col]<= Income_upper)
        elif col=='NumberCarsOwned':
            if value == 0:
                mask= (df[col]==0)
            elif value ==1:
                mask= (df[col]==1)
            else:
                mask= (df[col]> 1)
        elif col=='TotalChildren':
            if value == 0:
                mask= (df[col]==0)
            else:
                mask= (df[col]> 0)
        elif col=='NumberChildrenAtHome':
            if value == 0:
                mask= (df[col]==0)
            else:
                mask= (df[col]> 0)
        else:
            mask = df[col]==value
        df= df[mask]

    dfrows = len(df.index)    # number of rows in dataframe slice
    ones = df['BikeBuyer'].sum()        # number of bikebuyers in slice
    zeroes = dfrows - ones    # number of non-buyers in slice
    diff = ones-zeroes                  # difference
    flag = 1 if diff>0 else 0           # flag, 0/1 depending on ratio buyer/non-buyers
    #print df
    print i, flag, ones,zeroes, diff
```

CustomerExpenditureRegression.py

```
# Import modules
import pandas as pd
import numpy as np
from sklearn import preprocessing
import os

# Features used in the regression model:
# 'CustomerID','Education','Occupation','Gender','MaritalStatus','NumberCarsOwned',
# 'NumberChildrenAtHome','TotalChildren','HomeOwnerFlag','Age','IncomeNorm',
# 'BikeBuyer', 'AvgMonthSpend']

# Read Customers Database
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersDatabase002.csv'
pathfile= os.path.join(path,filename)
df = pd.read_csv(pathfile, sep=',', index_col=0)
df.reset_index(inplace=True, drop=True) # CustomerID is removed here

# Remove features not to be used in the regression model, such as 'BikeBuyer'
# Remove features ranked low in the feature_importance output
dropcols = ['BikeBuyer','HomeOwnerFlag', 'NumberCarsOwned', 'TotalChildren']
df.drop(dropcols, axis=1, inplace=True)
data=df

# Read dataset to estimate regression (AWTest-regression)
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersTest002.csv'
pathfile= os.path.join(path,filename)
df_reg = pd.read_csv(pathfile, sep=',', index_col=0)
df_reg.reset_index(inplace=True,drop=True)
dropcols = ['TotalChildren','HomeOwnerFlag', 'NumberCarsOwned']
df_reg.drop(dropcols, axis=1, inplace=True)
#df_reg

# Remove labels from X dataframe
y_data = data['AvgMonthSpend'].copy()
data.drop(['AvgMonthSpend'], axis=1, inplace=True)

# Prepare X dataframes for regression
dummies=['Education',
         'Occupation',
         'Gender',
         'MaritalStatus',
         'NumberChildrenAtHome']
         #'NumberCarsOwned' removed due to low importance
         #'HomeOwnerFlag', removed due to low importance
         #'TotalChildren', removed due to low importance
x_data = pd.get_dummies(data, columns=dummies)
x_reg = pd.get_dummies(df_reg, columns=dummies)
# Padd with Zeroes missing columns in x_reg
for col in x_data.columns:
    if col not in x_reg.columns:
        x_reg[col]=0

# Select a Data Normalization method with i
i=0
if i==0:
    scalername='None'
    xnorm= x_data
elif i ==2:
    scaler = preprocessing.MinMaxScaler().fit(x_data)
    xnorm=scaler.transform(x_data)
elif i ==3:
    scaler = preprocessing.StandardScaler().fit(x_data)
    xnorm=scaler.transform(x_data)
elif i==4:
    scaler = preprocessing.MaxAbsScaler().fit(x_data)
    xnorm=scaler.transform(x_data)

# Split data set into train and test
from sklearn.model_selection import train_test_split
(xtrain, xtest, ytrain, ytest) = train_test_split(xnorm, y_data, test_size=0.3, random_state=0)

# DECISION TREE==========
from sklearn import tree
model = tree.DecisionTreeRegressor(max_depth=7, random_state=None,splitter='best')
model.fit(xtrain,ytrain)
featureimp=pd.DataFrame(model.feature_importances_,index=x_data.columns,columns=['rank'])
topfeatures= featureimp.sort_values('rank', ascending=False)
DTscore=model.score(xtest,ytest)
DTpredict=model.predict(x_reg)

# RESULTS ==============
print "Decision Tree Score:{}".format(round((DTscore*100),3))
print "Features Rank:{}".format(topfeatures)
print "Decision Tree Prediction:{}".format(DTpredict)
```

CustomerExpenditureSlicer.py

```
# Code Description: This code slices the Customer´s Database (AWCustomers.csv) 25 times, once for each row
# in the AWTest-regression file. On each iteration, The AWCsutomers table is sliced with the value
# of each feature in row (i) of the For loop. When the AWCustomers table has been sliced, the mean
# 'AvgMonthSpend' in then calculated.

import pandas as pd
import numpy as np
import sklearn.metrics as metrics
import os


# read Customers Database
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersDatabase.csv'
pathfile= os.path.join(path,filename)
db = pd.read_csv(pathfile, sep=',', index_col=0)
db.reset_index(inplace=True, drop=True)
dropcols = ['AgeGroup', 'BikeBuyer']
db.drop(dropcols, axis=1, inplace=True)

# Read unseen dataset to Classify
path = (r'C:\Home\Ser\MOOCs\MSDataProg\Project\AWCustomers\Datasets')
filename = 'CustomersTest.csv'
pathfile= os.path.join(path,filename)
dfclass = pd.read_csv(pathfile, sep=',', index_col=0)
dfclass.reset_index(inplace=True,drop=True)
dropcols = ['AgeGroup']
dfclass.drop(dropcols, axis=1, inplace=True)
dfclass

#
for i in range(25):
    df=db
    headers = dfclass.columns
    for col in headers:
        value=dfclass.loc[i,col]
        if col=='Age':
            age_limit = 3 # number that determines range of age window
            age_lower = value - age_limit
            age_upper = value + age_limit
            mask= (df[col]>= age_lower) & (df[col]<= age_upper)
        elif col=='IncomeNorm':
            Income_limit = 1000 # number determining range of income window
            Income_lower = value - Income_limit
            Income_upper = value + Income_limit
            mask= (df[col]>= Income_lower) & (df[col]<= Income_upper)
        elif col=='NumberCarsOwned':
            if value == 0:
                mask= (df[col]==0)
            elif value ==1:
                mask= (df[col]==1)
            else:
                mask= (df[col]> 1)
        elif col=='TotalChildren':
            if value == 0:
                mask= (df[col]==0)
            else:
                mask= (df[col]> 0)
        elif col=='NumberChildrenAtHome':
            if value == 0:
                mask= (df[col]==0)
            else:
                mask= (df[col]> 0)
        else:
            mask = df[col]==value
        df= df[mask]

    dfrows = len(df.index)
    Average = df['AvgMonthSpend'].mean()
    print i, Average
```