a presentation on DataX’s analysis of the results of Office supplies limited’s telemarketing campaign results

December 5 2019.
Jehoram Mwila

DataX

Office supplies limited

OUTLINE

BACKGROUND

OFFICE SUPPLIES LIMITED

- Office supply limited recently conducted a telemarketing campaign on its existing customers and collected over 1600 entries of data.

- Using the results of the campaign, they would like to:
    - Generate list of customers that are more likely to respond to future campaigns and
    - What products they would likely buy.
    - Know what the cost of future campaigns targeted to those customers would be.
    - Find out what the profitability for each transaction would look like.

- They have engaged the services of DataX to develop a model that will predict the best possible leads for their sales team.

CAMPAIGN DETAILS

- The campaign featured randomly selected products Desks, Executive Chairs, Standard Chairs, Monitors, Printer Computers, Insurance, Toner and Office Supplies.

- Office supplies limited's telemarketing campaign cost and profit structure is as follows:
  - The company makes 22% on every sale as gross margin.
  - While it costs them $45.65 to contact every business
  - Plus an additional cost of $8.40 to complete every transaction.

- Calculated expected profit resulting from each transaction, using the formula:

E(profit) = $\boxed{.22 * \text{Prob(Sale)} * \text{Estimate(Transaction size)} - \$8.40 * \text{Prob(Sale)} - \$45.65}$

OBJECTIVES

OBJECTIVES

- ❖ To find those features of the campaign data that are important and have greatest effect. Then use those features to develop a model to predict which customers are likely to purchase a product during future similar campaigns

- ❖ To use the model to make forecast of profit estimates based on the campaign transaction costs and campaign sales revenue.

- ❖ To illustrate the contributions based on profitability of our model versus random sampling through a lift chart

BENEFITS TO OFFICE SUPPLY LIMITED

- ❖ This will help the company target right customers at reduced costs.
- ❖ Will help inform the company's decision on what products and the right quantities of those products to make sufficiently available to potential customers.
- ❖ To maximize the companies sales by targeting only potential customers during a future campaigns.

APPROACH

DATA PREPROCESSING

- ❖ Cleaned the data set by:
  - ➢ Changing some features from continuous variables to discrete variables.
  - ➢ Filled in missing values for some features and replaced NaNs.

- ❖ At this stage we obtained equal number of entries for each feature.
  - ➢ Proceeded to drop feature columns having less significance.

- ❖ Columns used in constructing models:

* Campaign period sales	* desk	* printer	* toner
* Customer number	* executive chair	* computer	* monitor
* Historical sales volume	* standard_chair	* insurance	* language
* Number of prior year transactions	* number_of_employees	* office_supplies	

- ❖ Added an extra column (purchase_or_not) to identify customers having campaign period sales greater than 0 as having made purchases with the rest as not.

SELECTING A DATA SET AND CONSTRUCTING MODELS

- Split the data 3:1 for training and validation set to build the models.

  - First was a **logistic regression** model (with purchase_or_not as the target variable) that predicts the probability that a customer will purchase a product during future similar campaigns.

  - Second was a **multivariate linear regression model** (used all the variables on page 11) to calculate the expected sales forecast of future campaigns sales from purchases made by customer identified in the first model.

ANALYSIS OF RESULTS

LOGISTIC REGRESSION VALIDATION TEST

LOGISTIC REGRESSION MODEL

```
[52]: logreg = LogisticRegression()
      logreg.fit(X_train, y_train)
      success_pred = logreg.predict(X_test)

/home/jehoram/anaconda/lib/python3.6/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

```
[53]: accuracy_score(y_true=y_test, y_pred=success_pred)

[53]: 0.8343224530168151
```

```
[54]: #Remember, for the classifier we can predict probabilities
      probs = logreg.predict_proba(X)
      probs[:5, :] # The left column is the probability of not success and right colum probability of success

[54]: array([[0.43484047, 0.56515953],
             [0.47830644, 0.52169356],
             [0.52748043, 0.47251957],
             [0.54885274, 0.45114726],
             [0.09995583, 0.90004417]])
```

```
[55]: probs.sum(axis=1)[:5] # The sum of the two probabilities is equal to 1

[55]: array([1., 1., 1., 1., 1.])
```

```
[56]: # logreg.predict vs logreg.predict_proba
      print(success_pred[:5]) #get the values of the first 5 success predictions
      print(probs[:5,1]) # get the firt 5 probabilities of the second column (success column)

      [0 0 0 0 0]
      [0.56515953 0.52169356 0.47251957 0.45114726 0.90004417]
```

Figure 1. Results of logistic regression

LOGISTIC REGRESSION VALIDATION TEST

- ❖ The accuracy score of our Logistic regression model as highlighted in the previous slide is 83%.

- ❖ This means that when using our model to predict the probability of a customer purchasing a product during future campaigns the model prediction is 83% correct.

- ❖ For every 100 customers our model predicts as potential purchasers during future campaigns 83 of them actually are.

MULTIVARIATE LINEAR REGRESSION VALIDATION TEST

❖ Multivariate linear regression model

```
LINEAR REGRESSION MODEL

[61]: # Now the linear regression model
      linreg = LinearRegression()
      linreg.fit(lin_X_train, lin_y_train)

[61]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

[62]: linreg.score(lin_X_test,lin_y_test)

[62]: 1.0

[66]: # df[['campaign_period_sales']]

[68]: sales_predicted = linreg.predict(df[['campaign_period_sales','customer_number','historical_sales_volume','number_of_prior_year_transactions','desk','e
      sales_predicted = np.where(sales_predicted>0, sales_predicted, 0)
      sales_predicted

[68]: array([8936.85, 7264.92, 7458.75, ...,    0.  ,    0.  ,    0.  ])
```

Figure 2. Results of linear regression

❖ From the result of our regression score we see that accuracy of our model in predicting accurate sales during future campaigns is **1 (100%)**.

❖ Meaning using our model office supplies limited would be able to predict future campaign sales for potential customers with **0** error.

LIFT CHART

LiftChart

DECILES	number_of_customers	range_of_historical_sales_volume	average_historical_sales_volume	sales_made_during_campaign	stdv_sales_made_during_campaign	future_campaigns_predicted_sales	profit_projections
0	161	8470045.65333	664784.9904	2904026.33682	1430.16743599918	2904026.33681894	206125.80253301
1	161	6967691.73333	288011.1742	325334.85501	285.27163893971	325334.855010394	-50632.04226705
2	161	4591757.76	296000.364	173344.422138	222.092876339895	173358.742138968	-65114.732189037
3	161	4068858.56	351493.3771	156088.923048	250.937535683597	156088.923048384	-66252.041951267
4	161	4577291.33333	390837.0774	115339.062752	180.116961984783	115458.396086435	-70019.252244822
5	161	4114171.42857	422108.7402	83309.1077429	169.55266563327	83309.1077435661	-72312.206803766
6	161	6378788.94171	475924.6581	60819.1413952	173.285785121733	61121.2680625635	-73539.495385009
7	161	5265240	622891.5988	35985.7330857	144.279105174741	35985.7330861514	-75368.636354703
8	161	5868134.13333	979299.6109	45120.8411429	210.318357413048	45542.6744763172	-75039.568610319
9	161	16458674.54	2185060.679	77277.3313333	314.503959953572	77843.8313333258	-73966.619942905
TOTALS	**1610**	-	-	**3976645.75447**	-	**3978069.86780505**	-416118.79321587

Figure 3. Lift Chart

RELATIONSHIP OF TECH PRODUCTS

```
[ ]:

[191]: tech_products = ['computer','monitor','printer','toner']
       tech_product_df = office_supply_df.loc[:, tech_products]

       fig, ax = plt.subplots(figsize=(9,6))
       sns.heatmap(tech_product_df.corr(), cmap='Blues', annot=True, ax=ax);
```

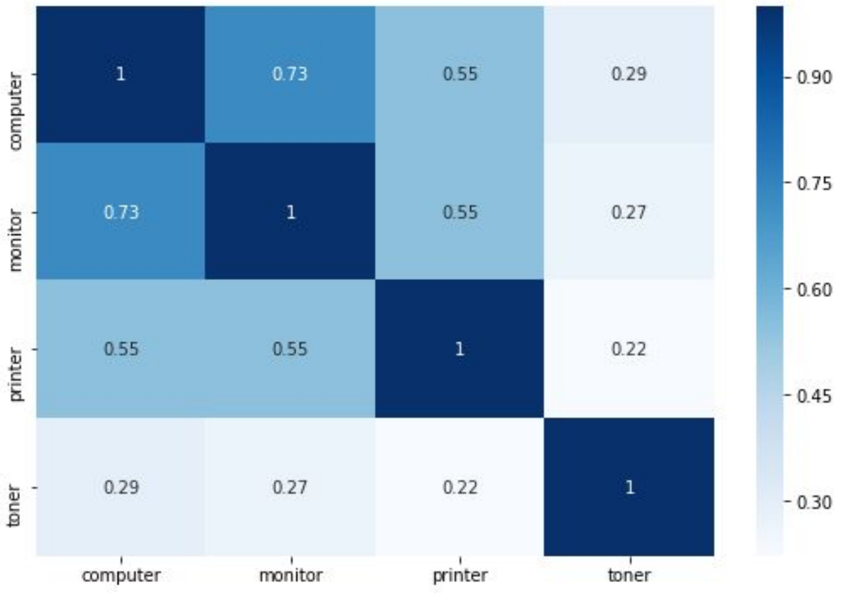

Figure 4. Tech products correlation result

RELATIONSHIP OF TECH PRODUCTS

- ❖ After examining the correlation heatmap of the 4 tech and tech-related products; computer, monitor, printer and toner on page 17, we observed that:

- ❖ Customers who buy computers as **highly** likely to buy monitors and the opposite is true.

- ❖ Customers that buy printers are likely to buy monitors and those that buy computers are also likely to buy printers and the opposite is true respectively.

RECOMMENDATIONS

RECOMMENDATIONS

- ❖ From the lift chart, for office supplies limited to be profitable during future campaigns they will have to target the customers in the first decile.
- ❖ We recommend that future telemarketing campaign should be targeted to the 161 customers in the decile 0.
- ❖ After further investigation we discovered that tech products (computer, monitor, printer and toner) performed much better than furniture products (executive chair, standard chair and desk) during the prior years transactions and in the recent campaign.
- ❖ While the sales of insurance and office supplies was lowest during both periods.
- ❖ We therefore recommend that office supplies limited acquires more tech products inventory to maximize their profits during future campaigns.

APPENDIX

RAW DATA SET VS DATA SET DURING PROCESSING

```
[2]: FILE_PATH = pathlib.Path.cwd().joinpath('Office_Supply_Campaign_Results.xlsx')
df = pd.read_excel(FILE_PATH)
print(df.info())
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16173 entries, 0 to 16172
Data columns (total 21 columns):
Customer Number                      16172 non-null float64
Campaign Period Sales                16172 non-null float64
Historical Sales Volume              16172 non-null float64
Date of First Purchase               16172 non-null datetime64[ns]
Number of Prior Year Transactions    16172 non-null float64
Do Not Direct Mail Solicit           16172 non-null float64
Do Not Email                         16172 non-null float64
Do Not Telemarket                    16172 non-null float64
Repurchase Method                    16172 non-null object
Last Transaction Channel             15730 non-null object
Desk                                 16173 non-null object
Executive Chair                      16171 non-null object
Standard Chair                       16171 non-null object
Monitor                              16171 non-null object
Printer                              16171 non-null object
Computer                             16172 non-null object
Insurance                            16170 non-null object
Toner                                16170 non-null object
Office Supplies                      16172 non-null object
Number of Employees                  16170 non-null object
Language                             11701 non-null object
dtypes: datetime64[ns](1), float64(7), object(13)
memory usage: 2.6+ MB
None
```

Figure 5. Raw Data

```
[232]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16173 entries, 5440 to 3530
Data columns (total 22 columns):
campaign_period_sales                16173 non-null float64
customer_number                      16173 non-null float64
historical_sales_volume              16173 non-null float64
number_of_prior_year_transactions    16173 non-null float64
repurchase_method                    16172 non-null object
last_transaction_channel             16173 non-null object
desk                                 16173 non-null int64
executive_chair                      16173 non-null int64
standard_chair                       16173 non-null int64
monitor                              16173 non-null int64
printer                              16173 non-null int64
computer                             16173 non-null int64
insurance                            16173 non-null int64
toner                                16173 non-null int64
office_supplies                      16173 non-null int64
number_of_employees                  16173 non-null float64
language                             16173 non-null int64
purchaser_or_not                     16173 non-null int64
probs                                16173 non-null float64
succ_pred                            16173 non-null int64
sales_predicted                      16173 non-null float64
profit_projections                   16173 non-null float64
dtypes: float64(8), int64(12), object(2)
memory usage: 2.8+ MB
```

Figure 6. Data during processing