Lead Scoring Case Study Summary

Group:

- Raghunath V P
- Raghavan Vinjamuri
- Sachin Singh

Summary

This document summarized the steps we incorporated in solving the lead scoring case study.

Before reading the data, we imported all the necessary modules required to solve the case study. In the python notebook, not all the imports are done at the start. Based on necessity of modules at each step, module imports are done.

1. **Importing the Data:**
   We import the data from gdrive/local folder (based on usage of colab or jupyter notebook) using pandas read_csv into a dataframe.

2. **Inspecting the dataframe:**
   Inspect the dataframe to check for any unnecessary headings, its shape, description and info.

3. **Data Preparation:**
   - First convert all the strings values to lower case. Then check the df to see if all the values are in lower case.
   - Replace ‘select’ values with np.nan.
   - Check for unique values using nunique. The nunique() method returns the number of unique values for each column.
   - Drop all the columns which has only one unique value.
   - Check for missing values and percentage of missing values.
   - Removing all the columns that have 35% or more null values
   - Dropping the Lead Number and Tags columns as it is not usable

- Even though "Specialization" is having more than 35% null values, we are not removing it as it is an important column. Removing "Specialization" can do damage in our analysis.
- Check again for percentage of missing values.
- "Country", "Specialization", "What is your current occupation", "What matters most to you in choosing a course" are still having lots of null values, let's not drop them as they are important fields. Fill all the null values with NA.
- Check again for the percentage of missing values.
- Check the total number of countries mentioned. Map all the countries to India, Outside India or NA.
- Check for the percent of lose if the null values are removed.
- Remove the prospect ID column as it is no more required for our analysis.

a. **Exploratory Data Analysis:**

   i. **Univariate analysis**

   Check for the data type of each column. For the categorical columns, use seaborn's countplot. For numerical columns, use histplot to analyse them.

   ii. **Bivariate analysis**

   Analyse categorical variables with respect to "converted" which is the target variable using countplot.

   Analyse all numerical variables using heatmap correlation.

b. **Dummy Variables for Categorical Variables:**

   Create dummy variables for categorical variables using get_dummies with drop_first as true. Next append this dataframe to main dataframe. Now remove original categorical attributes from the dataframe.

c. **Train – Test Split**

   "converted" is the target variable. Split the dataset into 70% and 30% for train and test respectively.

   Then scale the data using MinMax Scaler to make sure all the data is in between 0 and 1.

Now check for correlation among all the variables. This will help us when using RFE.

4. **Model Building:**

- Build a **logistic regression model**. Using RFE with 15 variables for feature selection, run RFE for that model. You will get a list of variables that have been selected by RFE.
- Next **assess the models** using statsmodels. Here, we will be checking the P-value and VIF values of variables. We will be considering 0.05 and 2.5 as the max values allowed for P-value and VIF respectively.
    - Max allowed value for Pvalue = 0.05
    - Max allowed value for VIF = 2.5
- You will get a stats models summary which gives us the p-value. Don’t forget to add constant before fitting.
- As you can see, the P value of "Last Notable Activity_had a phone conversation" and "What is your current occupation_housewife" is above 0.05.
- Before removing them, let’s check the VIF values for the variables.
- It can be observed that VIF values of all variables are < 2.5 but P-values for 2 variables are >0.05 showing their insignificance.
- Remove "Last Notable Activity_had a phone conversation".
- Refit the model with the new set of features.
- VIF of all variable seems fine but P-values for "What is your current occupation_housewife" > 0.05.
- Remove 'What is your current occupation_housewife'.
- Refit the model with the new set of features.
- **Next step is to predict on train data set.**
- Before predicting the probabilities on the train set, reshape y_pred to an array. Create a data frame with given convertion rate and predicted probabilities.

```
        Converted       Conversion_Prob
0       1               0.647883
1       0               0.133180
2       0               0.232946
3       0               0.133180
4       0               0.495090
```

- Create new column 'predicted' with 1 if Conversion_Prob > 0.5 else 0.

	Converted	Conversion_Prob	Predicted
0	1	0.647883	1
1	0	0.133180	0
2	0	0.232946	0
3	0	0.133180	0
4	0	0.495090	0

5. **Model Evaluation:**
   - Create confusion matrix using metrics.
   - Check for the overall accuracy.
   - **Accuracy** of the model is **81.02%**
   - Substitute the values of TP, TN, FP, FN.
   - Sensitivity = 69.58%
   - Specificity = 88.24%
   - When we are giving the cutoff as 0.5, we are getting
     * accuracy = 81.02 %
     * sensitivity ~ 70 %
     * specificity ~ 88 %
   - Previously, we chose a random cutoff at 0.05. To find the optimum value, we are plotting ROC Curve.
   - Call the ROC function. Let's create columns with different probability cutoffs. Now let's calculate accuracy sensitivity and specificity for various probability cutoffs. Let's plot accuracy sensitivity and specificity for various probabilities.
   - **From the graph, we can see that the optimum cutoff value is 0.35.**
   - When predicting with cutoff as 0.35, we can observe that we have accuracy, sensitivity and specificity around 80%.
   - **Next step is to predict on test dataset.**
   - Scale numeric values. Substitute all the columns in the final train model. Select the columns in X_train for X_test as well.
   - Make predictions on the test set and convert it to df
   - Convert y_test to dataframe.
   - Remove index for both dataframes to append them side by side.
   - Append y_test_df and y_pred_df.

- Rename column and Make prediction using cut off 0.35
- Check the overall accuracy. Accuracy = 80.79%.
- Create confusion matrix and Substitute the values of TP, TN, FP, FN
- Accuracy = 80.79%.
- Sensitivity = 81.30%
- Specificity = 80.50%
- With the cutoff value as 0.35, we have accuracy, sensitivity and specificity around 80%
- Next, we need to calculate Precision and Recall.
- With cutoff as 0.35, we have precision around 79% and recall around 70%.
- With cutoff at 0.41, we are having precision and recall around 75%.
- while making prediction using cut off 0.41, we have Precision around 73% and Recall around 76%.

6. **Conclusion:**

It was found that the variables that mattered the most in the potential buyers are:

1. The total time spend on the Website.
2. Total number of visits.
3. When the lead source was:
   a. Google
   b. Direct traffic
   c. Organic search
   d. Welingak website
4. When the last activity was:
   a. SMS
   b. Olark chat conversation
5. When the lead origin is Lead add format.
6. When their current occupation is working professional.

Keeping these in mind the X Education can flourish as they have a very high chance to get almost all the potential buyers to change their mind and buy their courses.

NOTE: We have only created two models and have large number of variables. It is not because of negligence or lack of knowledge but due to time constraint. We think that a model with variables around 7-9 which have VIF<2 for all variables will be a better one compared to our final model. Please consider the steps as we have followed everything.