Scholastic Travel Company (A)

Customer Retention Modeling

E2 - Group19
Amrithaa Yelagani, Filippo Schieratti, Rohan Ahuja, Tom Benabou, Mingxiang Xia

Executive Summary

Problem:
Scholastic Travel Company wants to build a Retention model to predict which consumers would book in the 2013-2014 school year, using historical data.

Dataset: Use the STC raw-data (stca.csv) from 2012-2013 consisting of 56 features and 2389 observations in order to train the model and do the prediction for the upcoming year.

Modelling: To predict the retention we use the column : “Retained.in.2012” which is used to determine if a customer will return or not. The column is labeled with 1/0. So, we use any of the “Classification Models” to train our data. Eg: Logistic Regression, Decision Tree, Random Forest, SVM, KNN etc.

Data Preprocessing: Null values treatment - Imputation and drop the columns that have too many missing values. Conduct EDA to identify which columns to be dropped and do some feature engineering. Identify highly correlated features and drop it accordingly.

Additional Methods: Treat the imbalance in the dataset and re-fit the model, which you further expand the scope of increasing the model accuracy.

Project Pipeline

Raw Data: (2389, 56)

Feature Engineering

Cleaned Data for Modelling
(2389, 26)

Training

Validation

Testing

Model Tuning
Logistic Regression & Decision Tree

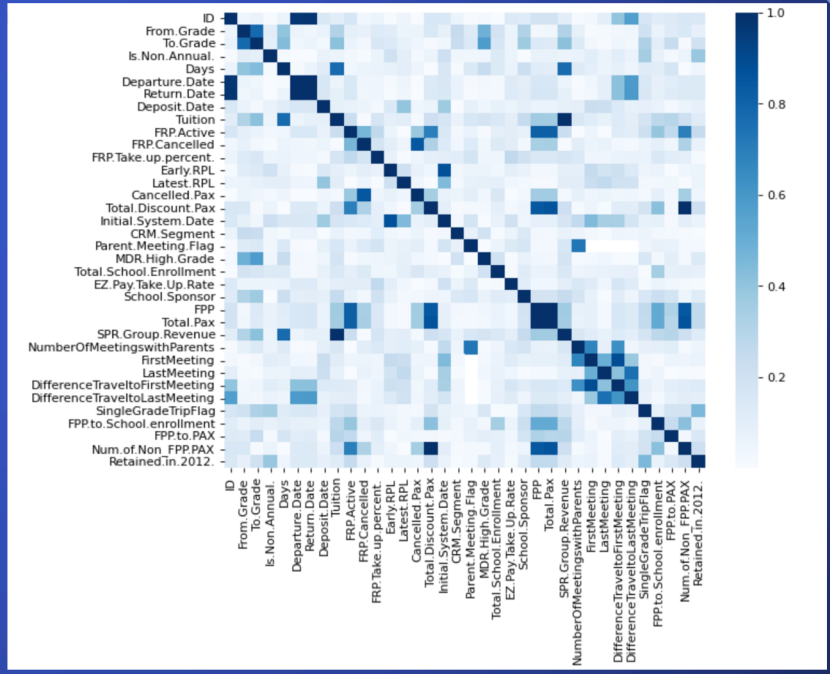
Evaluation
Using Confusion Matrix & Accuracy Scores

Final Model

Data Pre-Processing (1)

Index	Feature	Action	Comments
1	ID	drop	Unnecessary
3	from grade	drop	Redundant, kept To.grade (highest grade)
9	Departure.Date	drop	Unnecessary, we have the departure month data
10	Return.Date	drop	Unnecessary, we have the departure month data
11	Deposit.Date	drop	Unnecessary
12	Special.Pay	drop	Too many missing value and not relevant
14	FRP.Active	drop	Unnecessary, we have FRP.Take.up.percent
15	FRP.Cancelled	drop	Unnecessary, we have FRP.Take.up.percent
17	Early.RPL	drop	Unnecessary, redundant information
18	Latest.RPL	drop	Unnecessary, redundant information
19	Cancelled.Pax	drop	Unnecessary
20	Total.Discount.Pax	drop	Not relevant
22	Poverty.Code	drop	The same function as income level
23	Region	drop	Half of the customers fall in "others"
24	CRM.Segment	drop	Not interpretable and not relevant
26	Parent.Meeting.Flag	drop	Redundant, we have number of meetings
27	MDR.Low.Grade	drop	Redundant, kept TDR High Grade
29	Total.School.Enrollment	drop	Unnecessary, we have the school size indicator
31	EZ.Pay.Take.Up.Rate	drop	Not relevant
37	SPR.Group.Revenue	drop	Tuition and revenue is highly correlated
39	FirstMeeting	drop	Redundant, we have date from First Meeting
40	LastMeeting	drop	Redundant, we have date from First Meeting
42	DifferenceTraveltoLastMeeting	drop	Redundant, we have date from First Meeting
43	SchoolGradeTypeLow	drop	Redundant, grade information already reflected
44	SchoolGradeTypeHigh	drop	Redundant, grade information already reflected
45	SchoolGradeType	drop	Redundant, grade information already reflected
47	GroupGradeTypeLow	drop	Redundant, grade information already reflected
48	GroupGradeTypeHigh	drop	Redundant, grade information already reflected
50	MajorProgramCode	drop	Unnecessary and not relevant
53	FPP.to.PAX	drop	Unnecessary, we have FPP and PAX

Dropped the features in the table on the left, based on initial analysis of every feature and on correlation matrix. We dropped the columns that had correlation > 0.75



Data Pre-Processing (2)

1. Treating duplicates : No duplicates
2. Null values Imputation

```
# Replace missing values with the closet value
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=1)
df[['To.Grade']]=imputer.fit_transform(df[['To.Grade']])
df[['MDR.High.Grade']]=imputer.fit_transform(df[['MDR.High.Grade']])

# Replace missing values with the average of the column
df['Income.Level']=df['Income.Level'].fillna((df['Income.Level'].mean()))
df['DifferenceTraveltoFirstMeeting']=df['DifferenceTraveltoFirstMeeting'].fillna((df['DifferenceTraveltoFirstMeeting'].mean()))

# Replace missing values with the most frequent value
from sklearn.impute import SimpleImputer
imp=SimpleImputer(missing_values=np.nan, strategy="most_frequent")
df[['SchoolSizeIndicator']]=imp.fit_transform(df[['SchoolSizeIndicator']])
```

- To.Grade/MDR.High.Grade:
  150 values missing for “To.Grade” column and 68 missing for “MDR.High.Grade”, considering that the number should be integer, we could randomly assign the closest value to NA.

- Income.Level/DifferenceTraveltoFirstMeeting:
  The average value of the dataset can be calculated

- SchoolSizeIndicator:
  91 values missing, the value cannot be simply calculated, thus replace it with the most frequent value.

Feature Engineering

1. Convert all categorical variables to numerical

FEATURE	1	2	3	4	...
Program.Code	HS	HC	HD	SG	...
to grade	3	4	5	7	...
group state	CA	AZ	FL	MA	...
Travel.Type	A	B	T		
School.Type	PUBLIC	CHD	Catholic		
MDR.High.Grade	1	2	3	5	...
SPR.Product.Type	CA History	East Coast	Science	International	...
GroupGradeType	K->Elementary	Middle->Middle	High->High	Middle->High	...
SchoolSizeIndicator	L	M-L	S		

1. Feature Transformations & Interactions

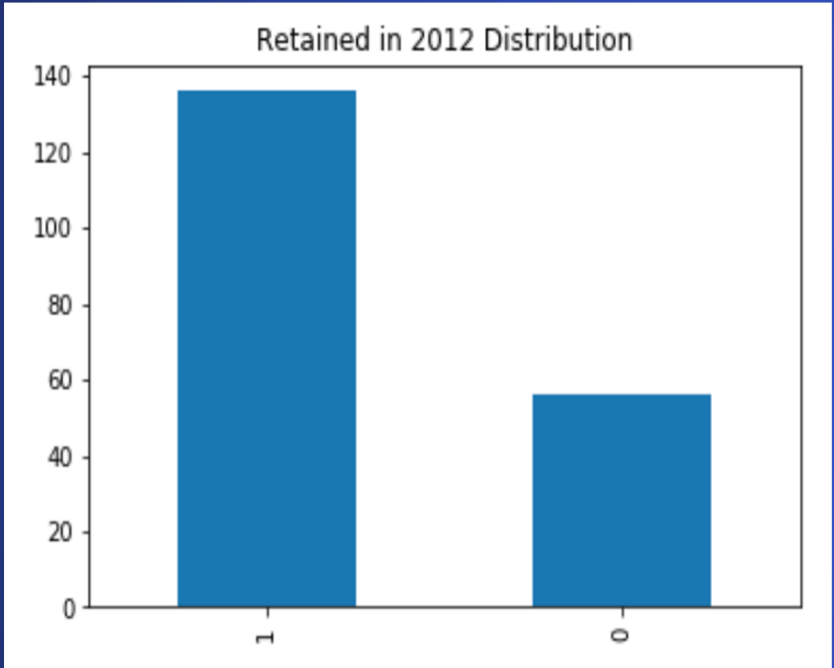
In order to capture more business insights, new features are formed by applying certain transforms such as:

Subtract "Departure.Date" from "Initial.System.Date", to get the length of period between the trip being organized to the departure date. If the period was too long, customers might lose interest and the overall satisfaction could drop, which could affect the future repurchase behavior.

Classification Modelling

LOGISTIC REGRESSION	DECISION TREE
Will the customer remain or not ? Is he/she my target customer ?	How to improve customer retention? (Model Explainability)
• Predicts the future repurchase behavior of a customer using the processed data. • Uses a particular threshold value to classify the likelihood of the probabilities into 1/0. Where, Class 1 = Retained Class 0 = Not-retained	• Predicts the retention but also gives the model explanation by giving the most important features • After knowing the most influential features, the ideal marketing strategy could be set forth in order to retain more customers.

Handling the Imbalanced Dataset



- In order to further increase the accuracy of the final model, treat the imbalance in the existing data set.
- Few techniques include Random Undersampling (or using Tomek Links) / Random Oversampling (or using SMOTE).