TELECOM CLIENT CHURN FORECAST USING MACHINE LEARNING

Chukwuemeka Okoli

7th January, 2022

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Summary of methodologies
  - Open the data and study the general information
  - Data Preprocessing
  - Exploratory Data Analysis
  - Modeling Process
  - Model Training
  - Model Analysis
  - Model Testing
- Summary of all results
  - Exploratory Data Analysis result
  - Predictive Analytics result

INTRODUCTION

- Project background and context

Interconnect telecom would like to be able to forecast their churn of clients. If it's discovered that a user is planning to leave, they will be offered promotional codes and special plan options. Interconnect's marketing team has collected some of their clientele's personal data, including information about their plans and contracts.

- Project Objectives

- Build a machine learning model to forecast Interconnect telecom's client churn

- Apply exploratory data analysis in determining whether special promotional services and plan options will discourage client churn

- Analyze the speed and quality of prediction, time required for training, etc.

Section 1

Methodology

OPEN THE DATA AND STUDY THE GENERAL INFORMATION

- The data consists of files obtained from different sources. By looking at the data, we find that:

  - “contract_data” has 7043 rows and 8 columns with no missing values and no duplicated values.

  - “internet_data” has 5517 rows and 8 columns with no missing values and no duplicated values.

  - “personal_data” has 7043 rows and 5 columns with no missing values and no duplicated values.

  - “phone_data” has 6361 rows and 2 columns with no missing values and no duplicated values.

DATA PREPROCESSING

- We merged the four dataset using the SQL-flavored merging in pandas

[8]:
```
# joining datasets
merged_df = pd.merge(contract_data, internet_data, on="customerID", how='left')
merged_df1 = pd.merge(merged_df, personal_data, on="customerID", how='left')
merged_df2 = pd.merge(merged_df1, phone_data, on="customerID", how='left')
merged_df2 = merged_df2.fillna('No')
merged_df2.sample(5)
```

[8]:

	customerID	BeginDate	EndDate	Type	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	InternetService	OnlineSecurity	Onlin
5613	3913-FCUUW	2014-02-01	No	Two year	Yes	Bank transfer (automatic)	70.45	5165.7	DSL	Yes	
6928	5306-BVTKJ	2016-02-01	No	One year	Yes	Credit card (automatic)	55.80	2651.2	DSL	No	
1761	5356-KZCKT	2015-04-01	No	Two year	Yes	Credit card (automatic)	24.45	1513.6	No	No	
6069	8097-OMULG	2015-03-01	No	One year	No	Credit card (automatic)	76.75	4541.9	DSL	No	
4624	0325-XBFAC	2019-05-01	2020-01-01 00:00:00	Month-to-month	Yes	Electronic check	94.70	740.3	Fiber optic	No	

- We replace column names and change data types
- We performed feature engineering to create “dayofweek”, “month”, “tenure”

EXPLORATORY DATA ANALYSIS

- We explored the data in order to generate insights from it.
- We tried to find out what payment type is unique to Interconnect's customers

[17]:
```
unique_payment_type_count = (telecom_df['type'].value_counts() / telecom_df['type'].value_counts().sum() * 100).tolist()

# unique payment type
unique_payment_type = telecom_df['type'].value_counts().reset_index().rename(columns={'index': 'type', 'type': 'unique count'})
unique_payment_type['percentage split (%)'] = ['{:.2f}'.format(x) for x in unique_payment_type_count]
unique_payment_type
```

[17]:

	type	unique count	percentage split (%)
0	Month-to-month	3875	55.02
1	Two year	1695	24.07
2	One year	1473	20.91

- We determined the payment methods that are unique to customer's

[18]:

	payment method	count	% payment split
0	Electronic check	2365	33.58
1	Mailed check	1612	22.89
2	Bank transfer (automatic)	1544	21.92
3	Credit card (automatic)	1522	21.61

EXPLORATORY DATA ANALYSIS

- We determined the services count by contract type

```
[21]: # services count grouped by contract type
(telecom_df.groupby('type', as_index=False)
     .agg({'service_count': 'sum'})
     .sort_values(by='service_count', ascending=False, ignore_index=True)
)
```

[21]:

	type	service_count
0	Month-to-month	6013
1	Two year	4654
2	One year	3686

- We determined if contract type affect customer churn

```
[27]: # effect of contract type on customer churn
contract_type_percent = telecom_df.groupby(
    'type', as_index=False).agg(
    {'exited': 'sum'}).sort_values(
    by='exited', ascending=False, ignore_index=True)
contract_type_effect = (telecom_df['type'].value_counts() / telecom_df['type'].value_counts().sum() * 100).tolist()
contract_type_percent['% exit percent'] = ['{:.2f}'.format(x) for x in contract_type_effect]
contract_type_percent
```

[27]:

	type	exited	% exit percent
0	Month-to-month	1655	55.02
1	One year	166	24.07
2	Two year	48	20.91

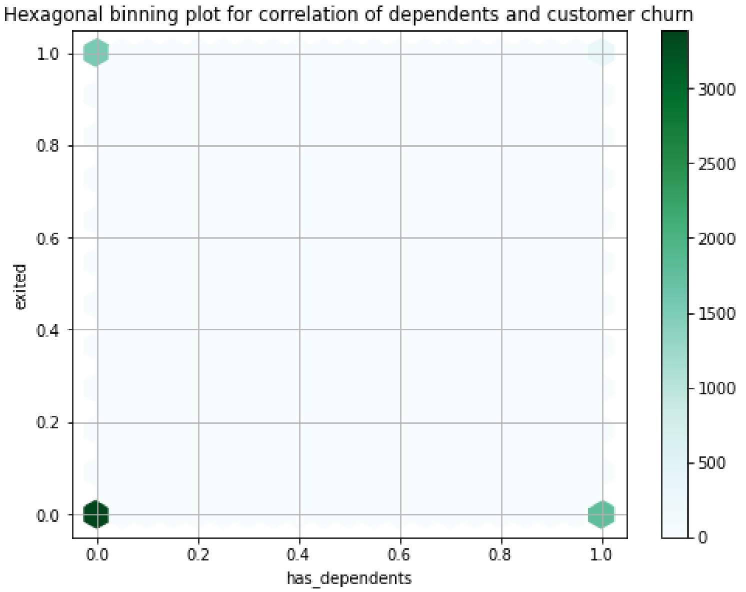
Section 2

INSIGHTS FROM EDA

EXPLORATORY DATA ANALYSIS

- Customers with less tenure are more likely to churn than well-established customers

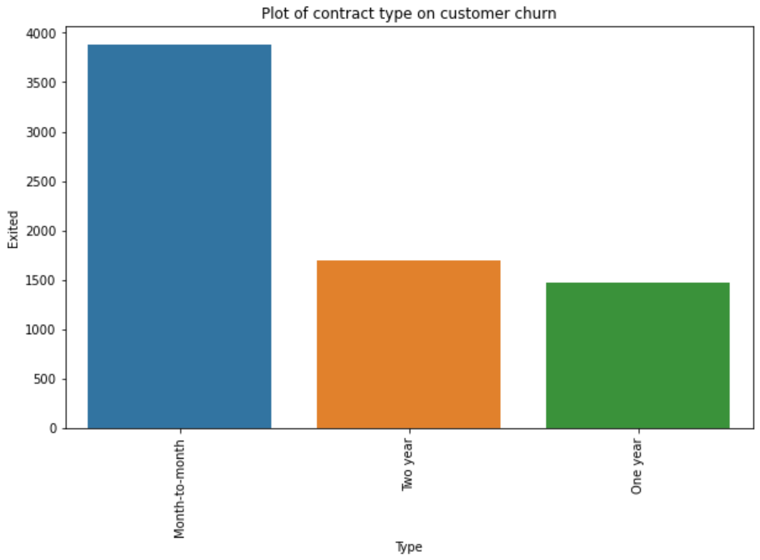
- Customers without dependents stayed longer with Interconnect telecoms than customers with dependent. It would make sense for Interconnect to target customers with less dependents.



EXPLORATORY DATA ANALYSIS

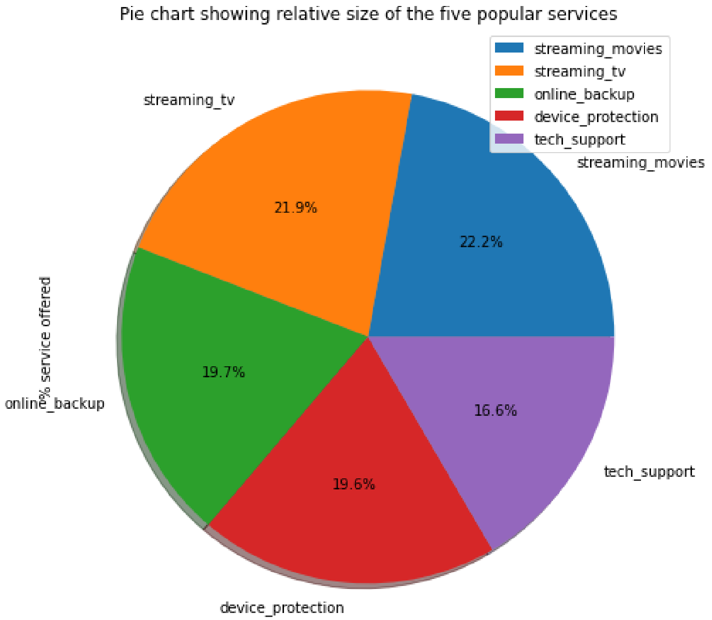
- Customers with two-year long contract tends to stay longer while customers on a month-to-month contract type churned faster.

```
[28]: # plot of contract type on customer churn
plot_snsbar(telecom_df, 'type', 'exited', 'Plot of contract type on customer churn')
```



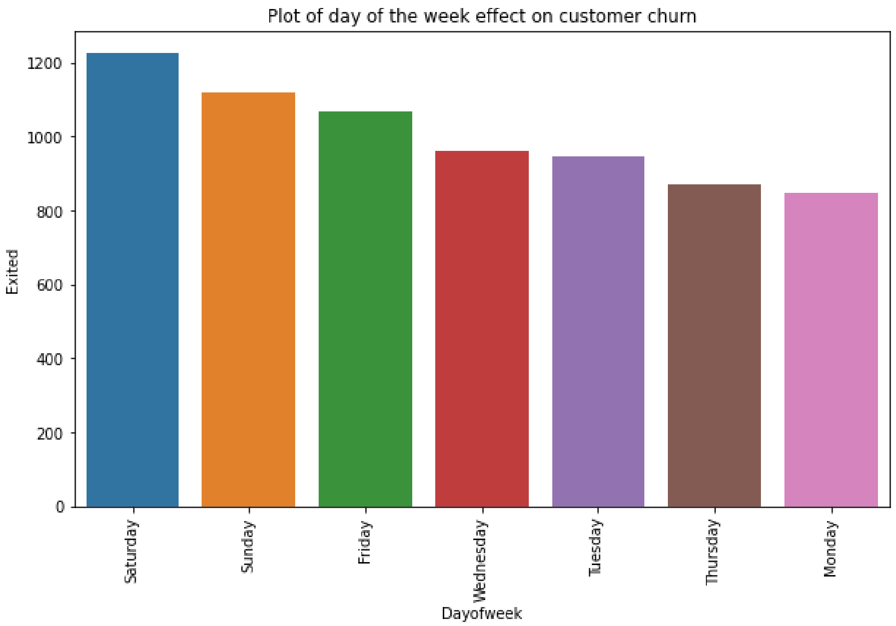
- The top 5 services offered are streaming movies, streaming tv, online backup, device protection and tech support.

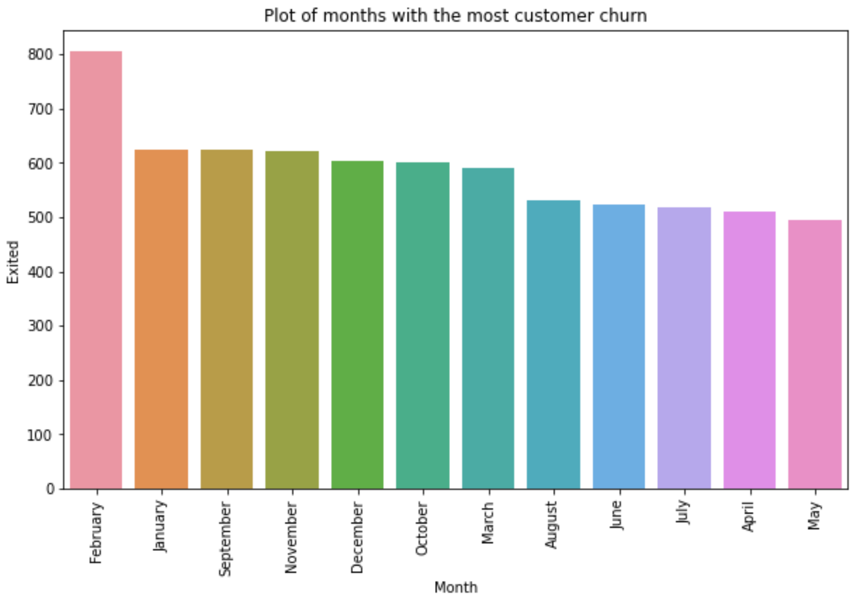
```
[31]: # plot of top 5 Interconnect service by count
telecom_services_pie = telecom_services.head(5)
(telecom_services_pie.set_index('services').plot(y='% service offered', kind='pie',
                        title = 'Pie chart showing relative size of the five popular services',
                        figsize=(8, 8), autopct='%1.1f%%', shadow=True)
);
```



EXPLORATORY DATA ANALYSIS

- Most churn occurred during the weekend.





- The months of February, September, November and December had the most churn.

INSIGHTS FROM EDA

- Most customers prefer month-to-month payment.
- Customers on a two-year contract bring in more revenue and churn less.
- Most churn occurs at weekend.
- Customers using more than 4 services churn less than others.

Action plan:

- Targeted marketing campaigns and promotional events should be done to promote the two-year plan to Interconnects customers.
- Encourage customers to make payment using electronic check.
- Introduce several bonuses, free service offerings for six months starting from September to February to prevent churn.
- Introduce special end of the week promotional events and services.

MODELING PROCESS

- The primary metric we used to evaluate the model is AUC-ROC. The secondary metric is accuracy.
- We performed feature engineering and encoded categorical variables using either one-hot encoding, label encoding or ordinal encoding.

Model type	Model	Encoding type	Highlight	Cons
Statistical based	Logistic regression	One-hot encoding	Less prone to over-fitting	Can overfit in high dimensional datasets
Tree-based	Decision Tree	label encoding	Normalization or scaling of data not needed	Prone to overfitting
	Random Forest	label encoding	Excellent predictive powers	Prone to overfitting
Gradient-boosted	Catboost	No encoding	Can handle categorical data well	Needs to build deep decision trees in features with high cardinality.
Gradient boosted	XGBoost	One-hot encoding	Good execution and model performance	Cannot handle categorical features (need encoding)
Gradient-boosted	LightGBM	Ordinal encoding	Extremely fast	Needs encoding for categorical features

- We split data into 75% training and 25% testing set
- We scaled the data by applying the standard scaler function
- We developed a baseline model

MODELING PROCESS

Baseline Model

```
[40]: # baseline model using a dummy classifier
dummy_clf = DummyClassifier(strategy="most_frequent")
dummy_clf.fit(features_train, y_train)
dummy_clf_test_predictions = dummy_clf.predict(features_test)
```

```
[41]: # evaluate baseline model
print_model_evaluation(y_test, dummy_clf_test_predictions)
```

```
F1 score:  0.000
Accuracy Score:  73.08%
Precision:  0.000
Recall:  0.000
Balanced Accuracy Score:  50.00%
AUC-ROC Score:  50.00%

Confusion Matrix
--------------------------------------------------
[[1287    0]
 [ 474    0]]

Classification report
--------------------------------------------------
              precision    recall  f1-score   support

           0       0.73      1.00      0.84      1287
           1       0.00      0.00      0.00       474

    accuracy                           0.73      1761
   macro avg       0.37      0.50      0.42      1761
weighted avg       0.53      0.73      0.62      1761
```

We developed a baseline model with accuracy of 73% and AUC-ROC score of 50%. This represents the baseline, so we expect our models to perform better.

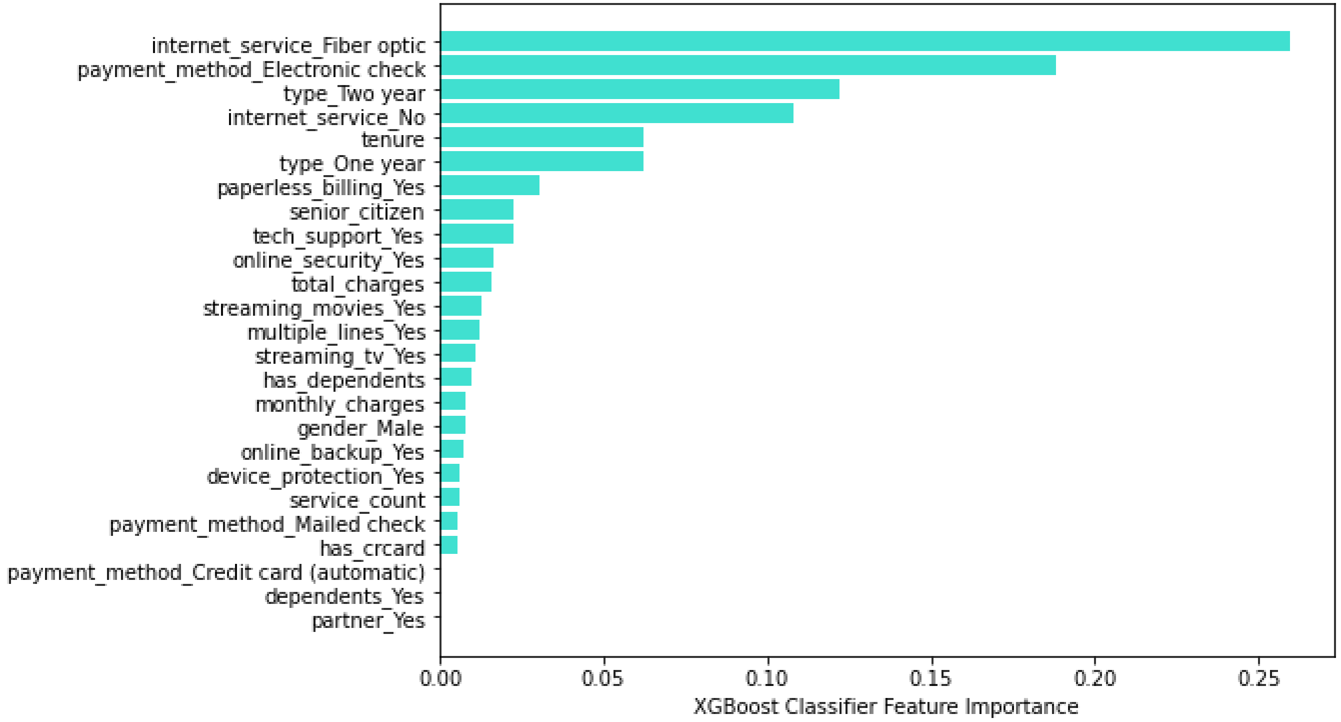
MODEL TRAINING

- We tuned hyperparameters for each model and cross validation during sampling of data for machine learning.
- We choose the best performing models on the training accuracy and AUC-ROC metric
- We also plotted the feature importance for each models
- We trained six models and chose the best performing model.
- The XGBoost classifier was the best performing model with a score of 91.3% on the training set.
- The XGBoost classifier was chosen as the model for the final testing on the test data because of its low hyperparameter tuning time, low prediction time and high accuracy.

MODEL TESTING

[78]:
```
#%%time
# make predictions with xgboost classifier for test data
xgboost_classifier_prediction(X_test_ohe, y_test_ohe)
```

```
AUC-ROC Score and Accuracy using XGBoost Classifier
AUC-ROC Score: 91.01%
Accuracy score: 87.90%
```



SUMMARY OF RESULTS

Models	Hyperparameter tuning time	Training time	Prediction time	AUC-ROC score	Accuracy score
Dummy Classifier	-	-	-	50.00 %	73.08 %
Logistic Regression	1.15 s	94.7 ms	422 ms	88.30 %	83.48 %
Decision Tree Classifier	11.3 s	41.9 ms	40.6 ms	87.87 %	84.44 %
Random Forest Classifier	1min 46s	173 ms	62.9 ms	88.89 %	84.04 %
CatBoost Classifier	21min 11s	15.4 s	67.9 ms	91.99 %	88.13 %
XGBoost Classifier	4min 16s	851 ms	161 ms	91.01 %	87.90 %
LightGBM Classifier	54.5 s	163 ms	101 ms	91.37 %	87.79 %

CONCLUSION

- We built a machine learning solution to forecast Interconnect telecom's client churn.
- We applied exploratory data analysis in determining whether special promotional services and plan option will discourage client churn.
- We analyzed the speed and quality of prediction, time required to train and prediction accuracy.

THANK YOU

Chukwuemeka Okoli