Predictive Analysis on Customer Churn

PAFBSC-401002 -- Group 4
Shizhao (Trixy) Xiong, Boyuan (Harry) Zhang, Xiaoyu (Rain) Zhou

Table of Contents

Our Business Problem

Increasing number of cancellations has become a notable problem since it cost much more to acquire new customers than to keep existing ones.

Our Solution

Use a churn prediction model to anticipate attrition by measuring the customer's propensity to churn, and then employ a customer retention marketing strategy to try to keep the customers.

Gather Data

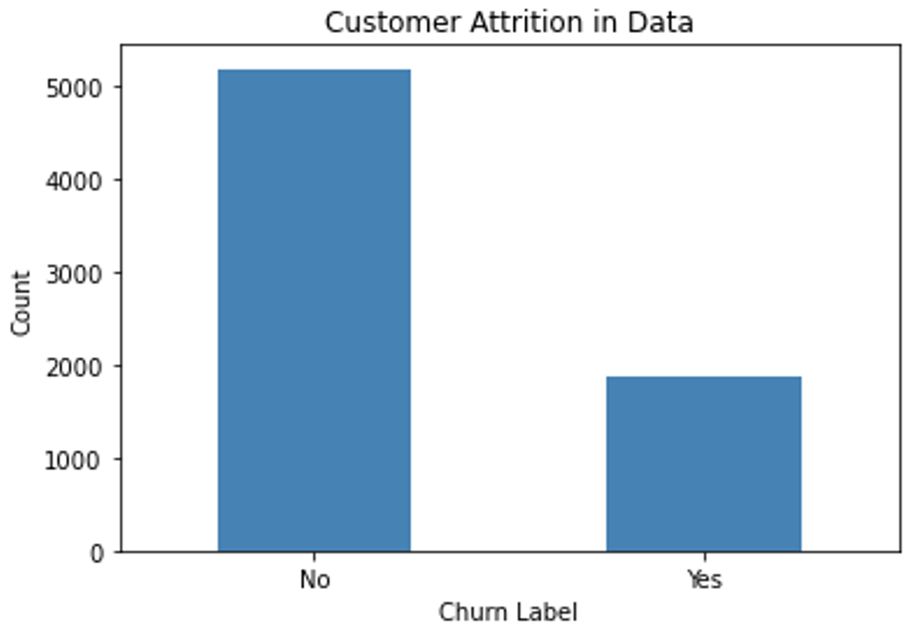
The data includes information about:

- Customers who left within the past month – the column is called churn
- Services that each customer has signed up for – phone services, multiple lines, internet services, online security, online backup, device protection, tech support, streaming TV, and streaming movies
- Customer account information – ID, tenure (number of months the customer has stayed with the company), contract, paperless billing, payment method, monthly charges, and total charges
- Demographic information about customers – gender, whether the customer is a senior citizen or not, and if they have partners and dependents

float64
int64
object

The dataset contains records of 7043 customers and 21 features.

Prepare Data - Customer Attrition in Data



churn customers: 1869

not churn customers: 5174

Overall churn rate is **26.54%**.

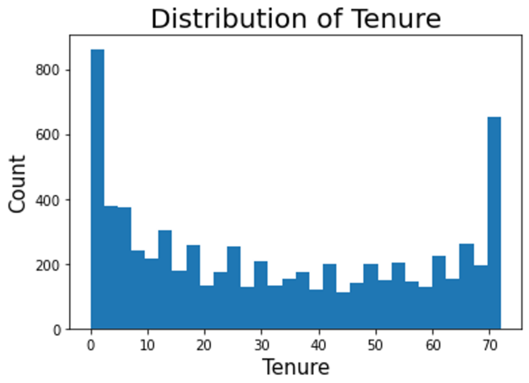
Prepare Data - Categorical Variable Distribution

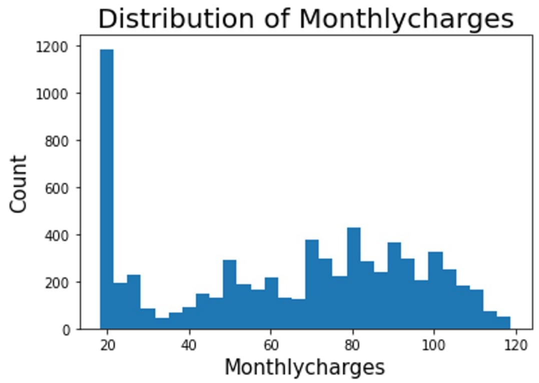
- We have almost equal number of both males and females.
- We have more young customers compared to senior.
- Customers with or without partners are about the same.
- We have more customers without dependent members than those who have.
- Most of the customers who don't have a phone service are way more than those who have.
- There is a common pattern in the features MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport. It is demonstrated that the majority of customers would rather not have access to these products than those who do.
- Customers who require internet service prefer DSL or Fiber optic.
- StreamingMovies and StreamingTV has similar plots, this suggests that there are an equal number of clients who prefer these services or do not.
- Customers prefer month-to-month contracts to other kinds, such as two-year or one-year contracts, in general.
- Most customers would rather have Paperless billing than any other form.

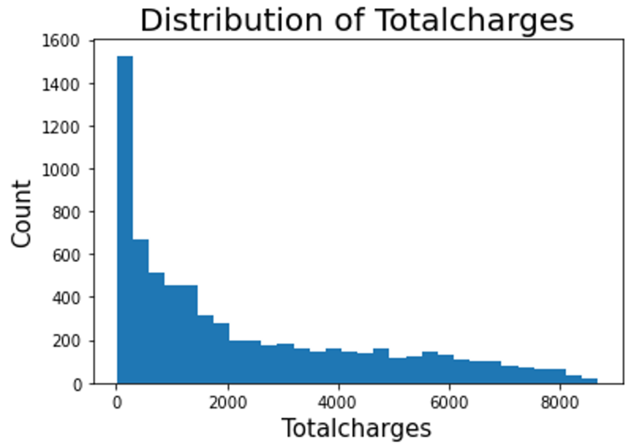
Prepare Data - Numeric Variable Distribution

- We see that many of our clients leave the company after less than ten months, another more significant increase occurs at more than tenure over 70 months.
- There are numerous tiny charges at around 20 dollars, other customers' monthly spending is more concentrated in the range of 70-100 dollars.
- We can set total charges to 0 because there are 11 missing values and only these 11 records have a value of 0 in the tenure (month of subscription) field, total charges is extremely near to monthly charges times tenure.
- Total Charge tends to decrease from 0 to 8000 dollars, especially in the 0-2000 range.

When compared to the remainder of their respective populations, many clients leave extremely early and with modest overall charges. Clients, on the other hand, begin to depart the organization when monthly prices approach 80 dollars.







Feature Engineering

Separate the entire data set into train and test data, which will allow you to save a hold-out set for final model evaluation, including not only the dataframes but also the indices.

Train size 5634 - **80%**

Test size 1409 - **20%**

2 new variable, "Non-Protection" and "TotalService”.
10 more ratio variables.
5 log of numerical fields.
720 statistical features.
Some polynomial which result in **15** new variables.

We now have **773** variables overall after the aforementioned stages.

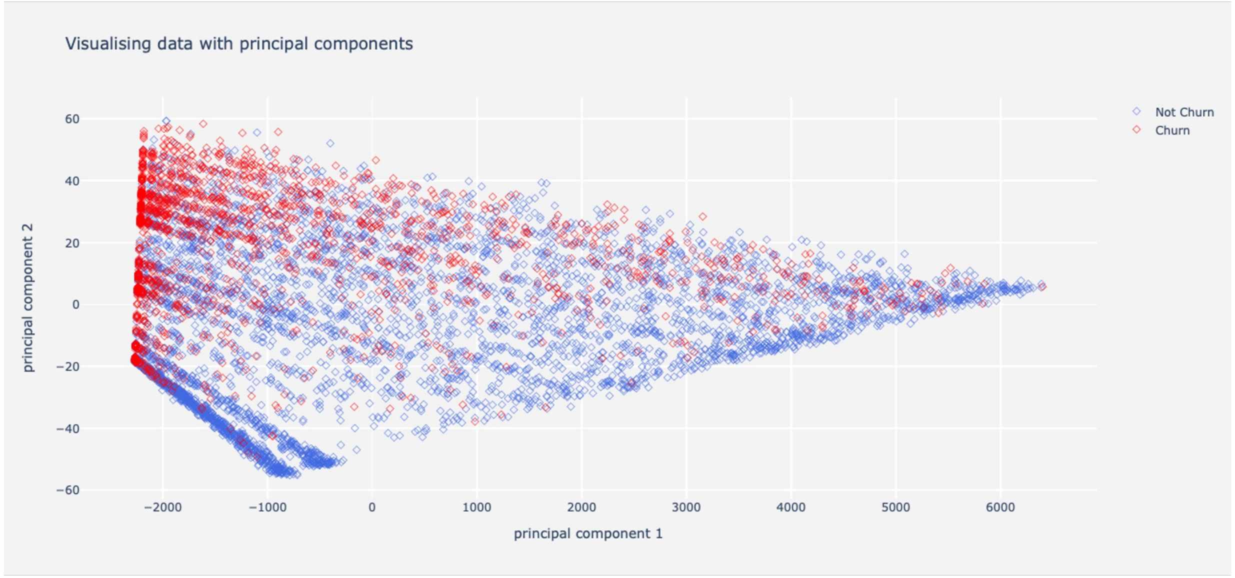
Feature Engineering - Visualising Data with PCA

Target Encoding

21 features

One-hot Encoding

32 features (since we create a new variable for each level of a categorical feature)



Feature Selection

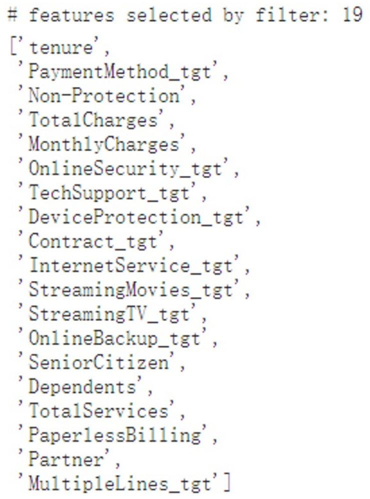
- Why?
    - Harder to fit nonlinear models as dimensionality increases.
    - Allows the consideration of many variables
    - Results in a sorted list so we know which variables to add in our model
- First use a filter: KS & Churn Detection Rate Univariate Filter
    - Fast
    - Looking at one variable at a time
    - How good is each variable by itself to predict y
- Then use a wrapper:
    - Multivariate
    - Remove correlations
    - Creates a short list of variables in the order of multivariate importance

KS & Churn Detection Rate Univariate Filter

- First define a function that calculate KS score and Churn Detection Rate for all features for both target encoded data and one-hot encoded data and store the result in a dataframe.
- Perform train test split on the encoded data
- Select the features with average score of KS and CDR above 0.2

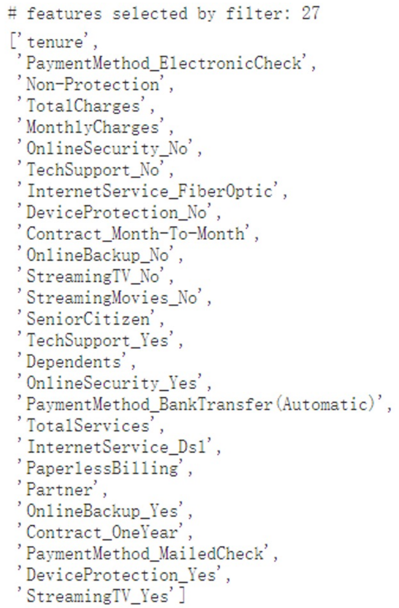
- Result:
- 19 features selected by filter for the target encoded data
- 27 features selected by filter for the one-hot encoded data

Target encoded:



```
# features selected by filter: 19
['tenure',
 'PaymentMethod_tgt',
 'Non-Protection',
 'TotalCharges',
 'MonthlyCharges',
 'OnlineSecurity_tgt',
 'TechSupport_tgt',
 'DeviceProtection_tgt',
 'Contract_tgt',
 'InternetService_tgt',
 'StreamingMovies_tgt',
 'StreamingTV_tgt',
 'OnlineBackup_tgt',
 'SeniorCitizen',
 'Dependents',
 'TotalServices',
 'PaperlessBilling',
 'Partner',
 'MultipleLines_tgt']
```

One-hot encoded:



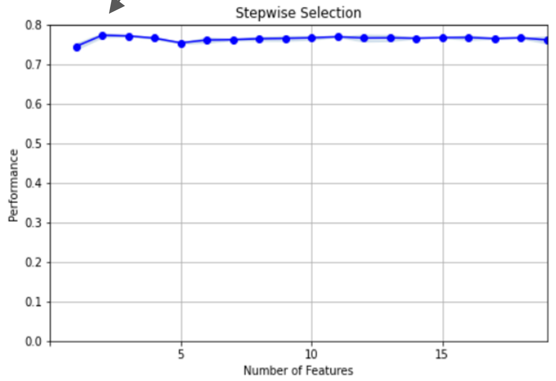
```
# features selected by filter: 27
['tenure',
 'PaymentMethod_ElectronicCheck',
 'Non-Protection',
 'TotalCharges',
 'MonthlyCharges',
 'OnlineSecurity_No',
 'TechSupport_No',
 'InternetService_FiberOptic',
 'DeviceProtection_No',
 'Contract_Month-To-Month',
 'OnlineBackup_No',
 'StreamingTV_No',
 'StreamingMovies_No',
 'SeniorCitizen',
 'TechSupport_Yes',
 'Dependents',
 'OnlineSecurity_Yes',
 'PaymentMethod_BankTransfer(Automatic)',
 'TotalServices',
 'InternetService_Dsl',
 'PaperlessBilling',
 'Partner',
 'OnlineBackup_Yes',
 'Contract_OneYear',
 'PaymentMethod_MailedCheck',
 'DeviceProtection_Yes',
 'StreamingTV_Yes']
```

Stepwise selection wrapper

- Goal: Create a short but conservative list of variables in the order of multivariate importance
- Correlations taken into account since one strong variable is chosen, another variable that is highly correlated with that variable is not going to add much to the performance.
- Allow to test on various number of input variables to see how the performance changes

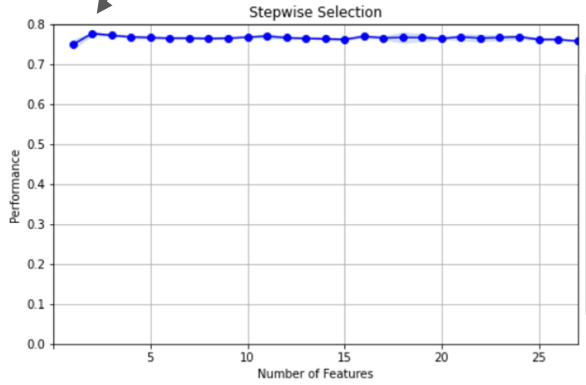
- Result:
- 10 features for the target encoded data
- 10 features for the one-hot encoded data

Target encoded:



['tenure',
'PaymentMethod_tgt',
'MonthlyCharges',
'Contract_tgt',
'InternetService_tgt',
'StreamingMovies_tgt',
'StreamingTV_tgt',
'Dependents',
'PaperlessBilling',
'Partner']

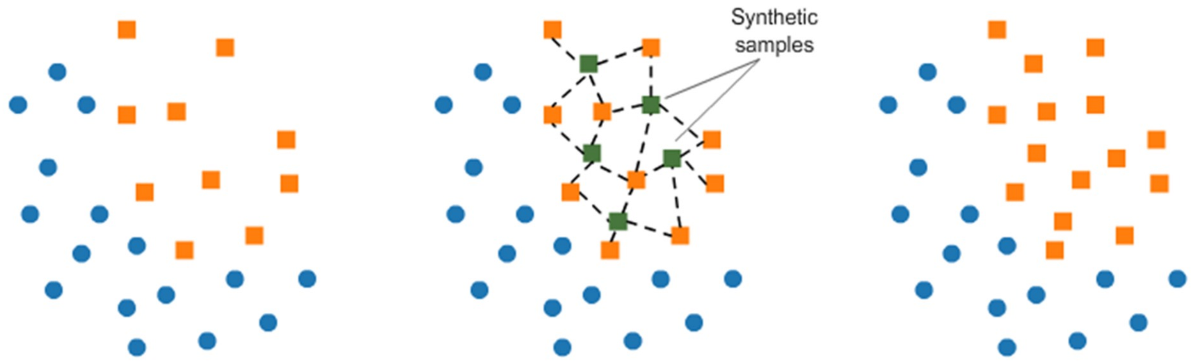
One-hot encoded:



['tenure',
'OnlineSecurity_No',
'TechSupport_No',
'InternetService_FiberOptic',
'Contract_Month-To-Month',
'StreamingTV_No',
'OnlineSecurity_Yes',
'PaperlessBilling',
'Contract_OneYear',
'StreamingTV_Yes']

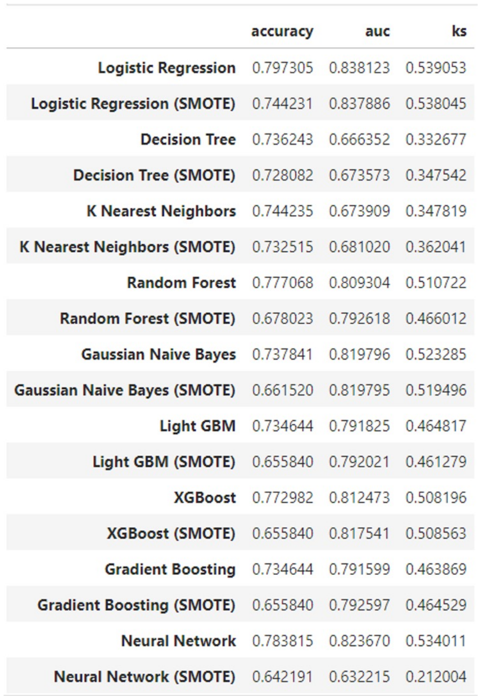
Model Building(Baseline)

- Using Synthetic Minority Oversampling Technique (SMOTE) to balance the two populations during model building
- A kind of technique that upsample (oversample) the bads (rare types)
- Randomly pick a point from the minority class.
- Compute the k-nearest neighbors (for some pre-specified k) for this point.
- Add k new points somewhere between the chosen point and each of its neighbors



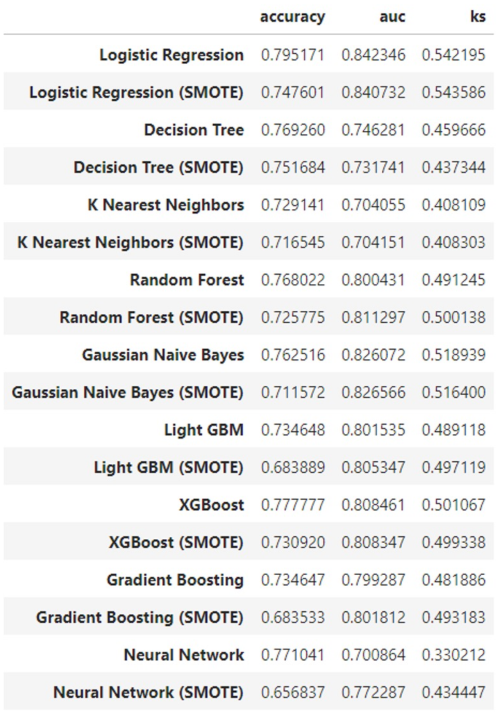
Algorithms used to build models

- A total of 9 algorithms used are logistic regression, decision tree, K Nearest Neighbors, Random Forest, and so on.
- Calculate average k-fold cross-validated accuracy, roc-auc, and KS scores.
- If smote is True, each model will be additionally trained on upsampled train sets during cross validation.
- Generate a summary dataframe



	accuracy	auc	ks
Logistic Regression	0.797305	0.838123	0.539053
Logistic Regression (SMOTE)	0.744231	0.837886	0.538045
Decision Tree	0.736243	0.666352	0.332677
Decision Tree (SMOTE)	0.728082	0.673573	0.347542
K Nearest Neighbors	0.744235	0.673909	0.347819
K Nearest Neighbors (SMOTE)	0.732515	0.681020	0.362041
Random Forest	0.777068	0.809304	0.510722
Random Forest (SMOTE)	0.678023	0.792618	0.466012
Gaussian Naive Bayes	0.737841	0.819796	0.523285
Gaussian Naive Bayes (SMOTE)	0.661520	0.819795	0.519496
Light GBM	0.734644	0.791825	0.464817
Light GBM (SMOTE)	0.655840	0.792021	0.461279
XGBoost	0.772982	0.812473	0.508196
XGBoost (SMOTE)	0.655840	0.817541	0.508563
Gradient Boosting	0.734644	0.791599	0.463869
Gradient Boosting (SMOTE)	0.655840	0.792597	0.464529
Neural Network	0.783815	0.823670	0.534011
Neural Network (SMOTE)	0.642191	0.632215	0.212004

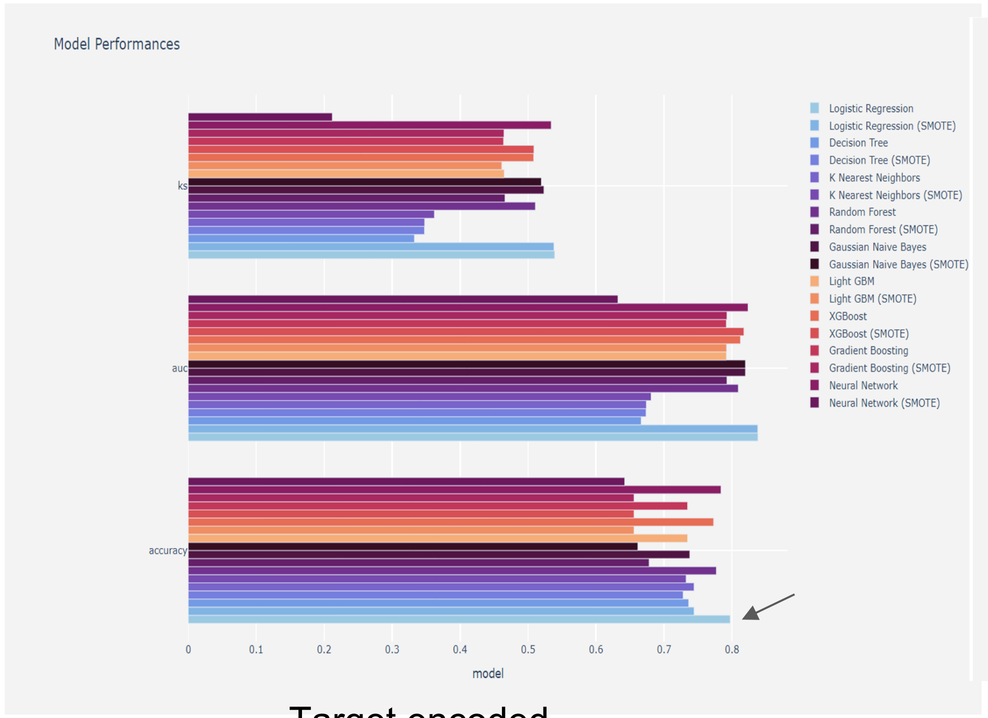
Target encoded



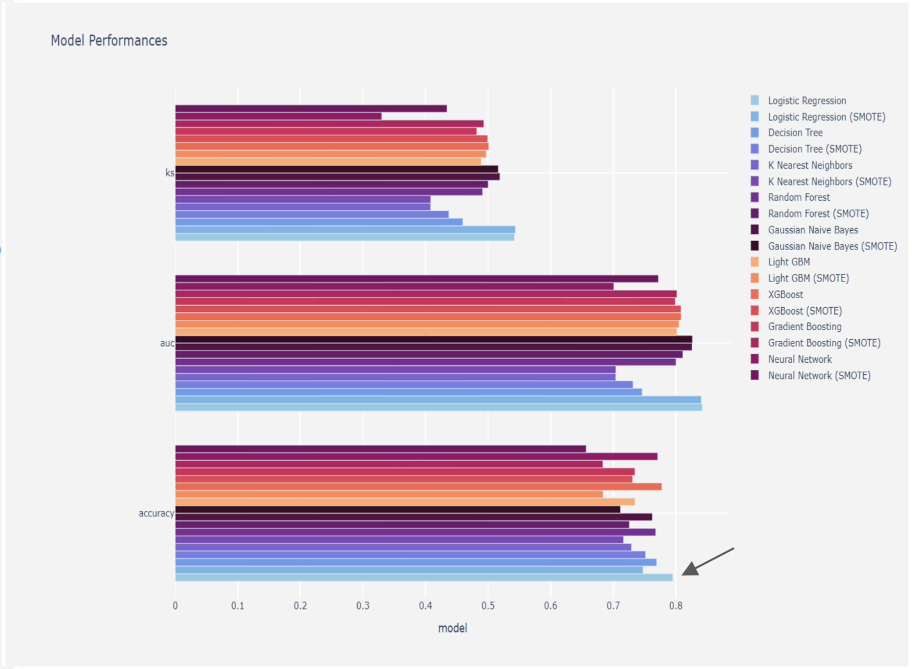
	accuracy	auc	ks
Logistic Regression	0.795171	0.842346	0.542195
Logistic Regression (SMOTE)	0.747601	0.840732	0.543586
Decision Tree	0.769260	0.746281	0.459666
Decision Tree (SMOTE)	0.751684	0.731741	0.437344
K Nearest Neighbors	0.729141	0.704055	0.408109
K Nearest Neighbors (SMOTE)	0.716545	0.704151	0.408303
Random Forest	0.768022	0.800431	0.491245
Random Forest (SMOTE)	0.725775	0.811297	0.500138
Gaussian Naive Bayes	0.762516	0.826072	0.518939
Gaussian Naive Bayes (SMOTE)	0.711572	0.826566	0.516400
Light GBM	0.734648	0.801535	0.489118
Light GBM (SMOTE)	0.683889	0.805347	0.497119
XGBoost	0.777777	0.808461	0.501067
XGBoost (SMOTE)	0.730920	0.808347	0.499338
Gradient Boosting	0.734647	0.799287	0.481886
Gradient Boosting (SMOTE)	0.683533	0.801812	0.493183
Neural Network	0.771041	0.700864	0.330212
Neural Network (SMOTE)	0.656837	0.772287	0.434447

One-hot encoded

Model Performances Comparison



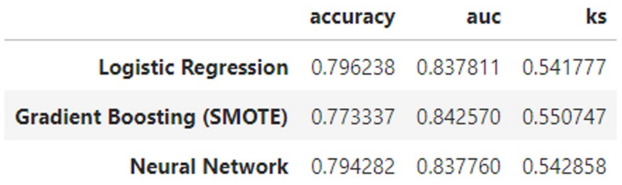
Target encoded



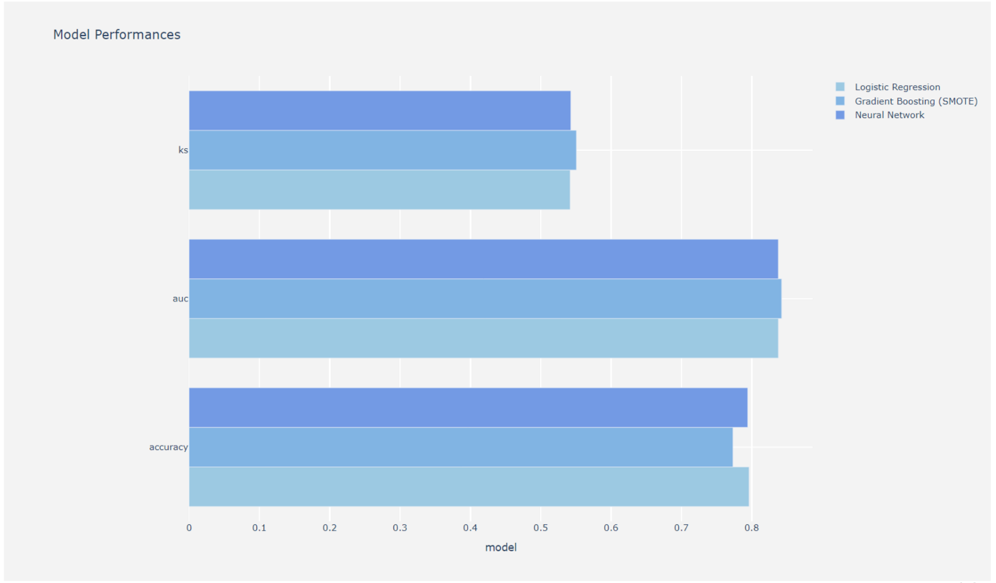
One-hot encoded

Model Tuning

- Process of maximizing model performance without overfitting or creating too high of a variance
- Accomplished by selecting the appropriate "hyperparameters"



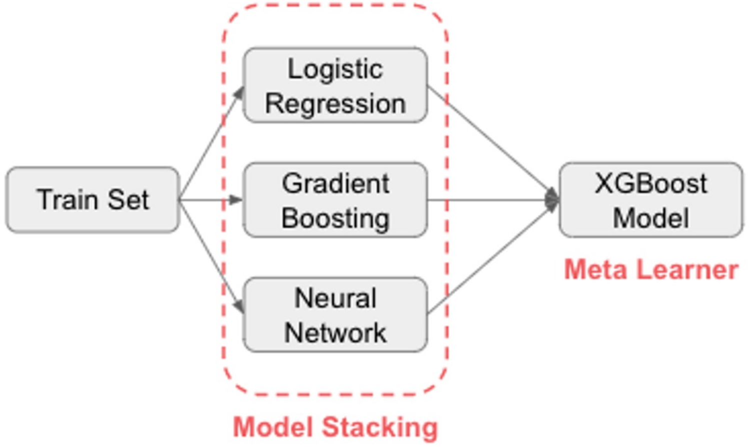
	accuracy	auc	ks
Logistic Regression	0.796238	0.837811	0.541777
Gradient Boosting (SMOTE)	0.773337	0.842570	0.550747
Neural Network	0.794282	0.837760	0.542858



Model Stacking/Ensembling

- **Model Stacking**: Potentially improve model prediction accuracy.
- Combined three individual base models (Logistic Regression, Gradient Boosting, Neural Network Models) and formed XGBoost Model by learning all the predictions the base models did.
- Stacking Model outperformed Neural Network model completely. However, compared to Logistic Regression and Gradient Boosting, XGBoost has lower performance for certain performance indicators.

In Predicting Customer Churn (train set),
Stacking Model (XGBoost) ≈ Gradient Boosting ≈ Logistic Regression > Neural Network



	Accuracy	AUC	KS
Logistic Regression	0.796	0.838	0.542
Gradient Boosting (SMOTE)	0.773	0.843	0.551
Neural Network	0.794	0.838	0.543
Stacking Model (XGB)	0.795	0.841	0.548

Model Evaluation on Hold-Out Set

- Test prediction performance of models we trained on hold-out set.
- For hold-out set only, **Logistic Regression model**, with a 80.3% prediction accuracy, outperformed all other models based on performance indicators.
- Both base models and stacking model have similar performance on hold-out set as on train set → no sign of overfitting.

Model Performance on Hold-Out Set

	Accuracy	AUC	KS
Logistic Regression	0.803	0.833	0.506
Gradient Boosting (SMOTE)	0.770	0.840	0.526
Neural Network	0.794	0.833	0.506
Stacking Model (XGB)	0.798	0.831	0.499

Models are trained well and will stay at the same accuracy when used to make predictions on new data.

In predicting customer churn (hold-out set):

Logistic Regression > Neural Network ≈ Gradient Boosting > Stacking Model (XGBoost)

Train Set vs. Hold-Out Set

	Accuracy		AUC		KS	
	Train	Hold-out	Train	Hold-out	Train	Hold-out
Logistic Regression	0.796	0.803	0.838	0.833	0.542	0.506
Gradient Boosting (SMOTE)	0.773	0.770	0.843	0.840	0.551	0.526
Neural Network	0.794	0.794	0.838	0.833	0.543	0.506
Stacking Model (XGB)	0.795	0.798	0.841	0.831	0.548	0.499

Insights + Recommendations

- **Logistic Regression** model has the overall higher performance because of its simplicity and high accuracy of prediction
  - Recommendation 1: select this model for future customer churn prediction.
- Formed the list of likely-to-churn customers as the target for the marketing retention program to prevent them from cancelling.
  - Recommendation 2:
    - Further customize marketing efforts for this target list by conducting **cluster analysis** to create segmentations
    - For each segment, customize **advertising content, media placements**, etc. for different segments to target customers more effectively.

OOH Ad

You're the best in your field. Now come be the best in ours.
At Verizon, we're making great leaps forward in Technology. At the same time, we're committed to helping the brightest women behind our groundbreaking work go further too.
Discover how we can help you advance by visiting verizon.com/TechnologyNetwork
verizon

Print Ad

THANK YOU!