IBM Telco Churn Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            7043 non-null   object
 1   seniorcitizen     7043 non-null   object
 2   partner           7043 non-null   object
 3   dependents        7043 non-null   object
 4   tenure            7043 non-null   int64
 5   phoneservice      7043 non-null   object
 6   multiplelines     7043 non-null   object
 7   internetservice   7043 non-null   object
 8   onlinesecurity    7043 non-null   object
 9   onlinebackup      7043 non-null   object
 10  deviceprotection  7043 non-null   object
 11  techsupport       7043 non-null   object
 12  streamingtv       7043 non-null   object
 13  streamingmovies   7043 non-null   object
 14  contract          7043 non-null   object
 15  paperlessbilling  7043 non-null   object
 16  paymentmethod     7043 non-null   object
 17  monthlycharges    7043 non-null   float64
 18  totalcharges      7043 non-null   float32
 19  churn             7043 non-null   object
dtypes: float32(1), float64(1), int64(1), object(17)
memory usage: 1.0+ MB
```

The dataset we have chosen is the IBM telecom churn dataset available from Kaggle.

The data set contains 7043 observations and 21 columns with 20 variables.

Problem Motivation

- The telco sector is one of the most competitive industry, and in order to survive, telco companies have to maximise revenue (highest ROI to increase customer retention)
- Customer churn is a major problem and one of the most important concerns for large companies and is the main reason why understanding customers is important.

Problem Definition

- Can we predict whether customers will churn?
- Can telco customers be clustered into different segments to help the telco understand their customer segments and create targeted strategies?

Data Cleaning

CustomerID

Variable that was dropped from the dataset

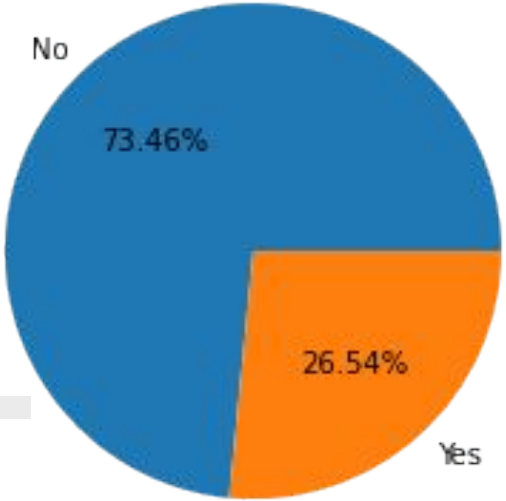
	customerID
0	7590-VHVEG
1	5575-GNVDE
2	3668-QPYBK
3	7795-CFOCW
4	9237-HQITU

TotalCharges

The Null Values were filled in using the formula monthlycharges*tenure = totalcharges

	Contract	tenure	MonthlyCharges	TotalCharges
1340	Two year	0	56.05	
936	Two year	0	80.85	
6670	Two year	0	73.35	
4380	Two year	0	20.00	
5218	One year	0	19.70	
...	...	...	...	...
914	Two year	72	25.20	1798.9
917	Two year	72	65.55	4807.45
4574	One year	72	105.75	7629.85
3635	Two year	72	24.55	1750.7
3543	Two year	72	105.60	7581.5

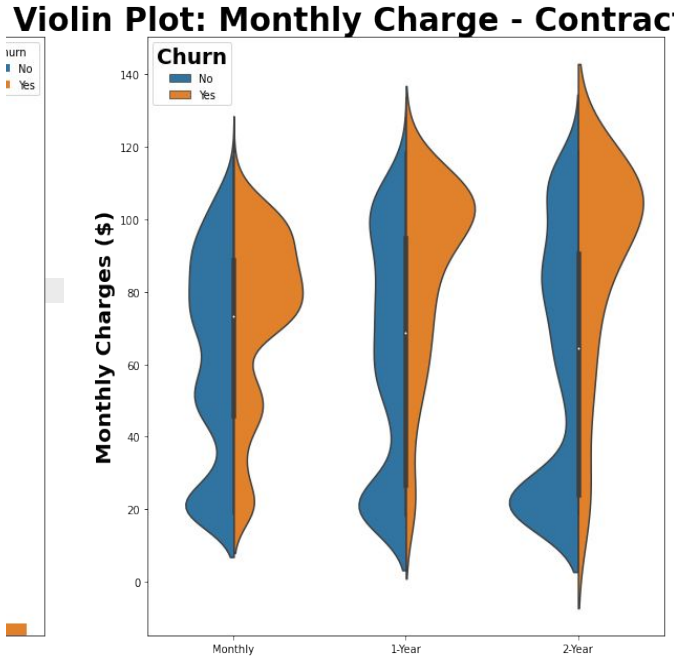
Churn Ratio of the Telco



1. 26.54% of the customers have stopped using the service in a span of 3 years or less

2. Based on this variable we aim to create and get telcos to adopt better marketing strategies for the customers.

Exploratory Data Analysis

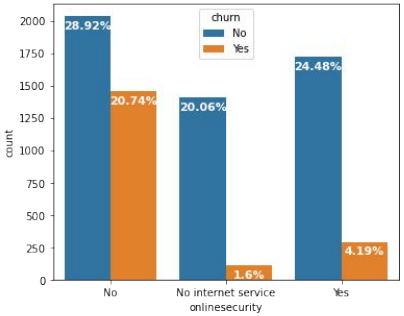


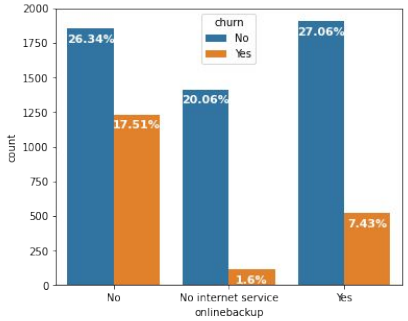
- **Contract Types:** Monthly, One-Year, Two-Year
- More customers churn when on monthly contracts with average costs of more than $60/month as we can see on the violin plot
- As contract durations increase, churn decrease showing telcos should market for longer tenures

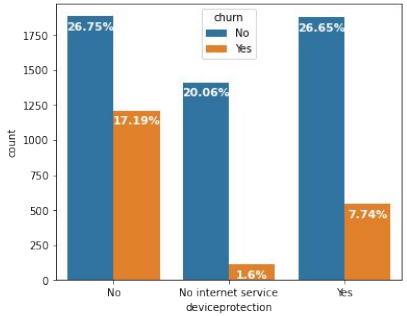
Monthly Charges: Indicates the customer's current total monthly charge for all their services from the company.

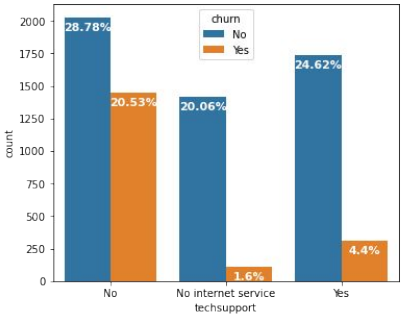
Tenure in Months: Indicates the total amount of months that the customer has been with the company by the end of the quarter specified above.

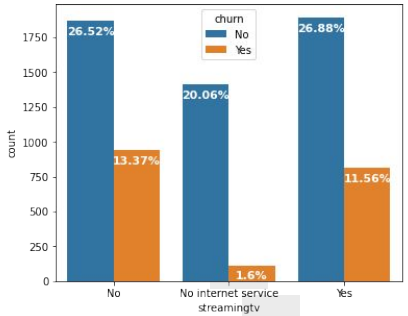
Add - on Services

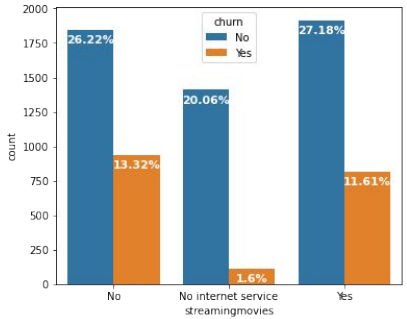








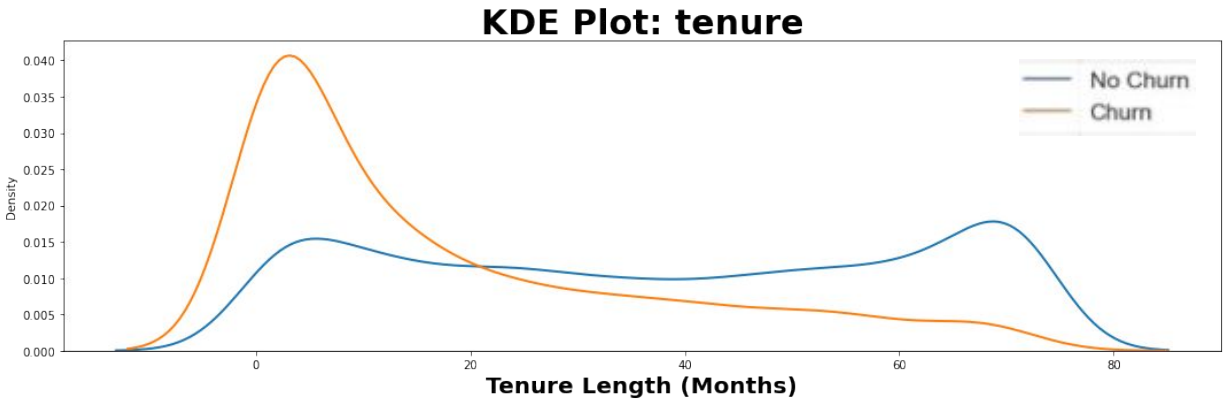




- Customers with online security and/or tech support churn the least and are ready to pay for the extra service available

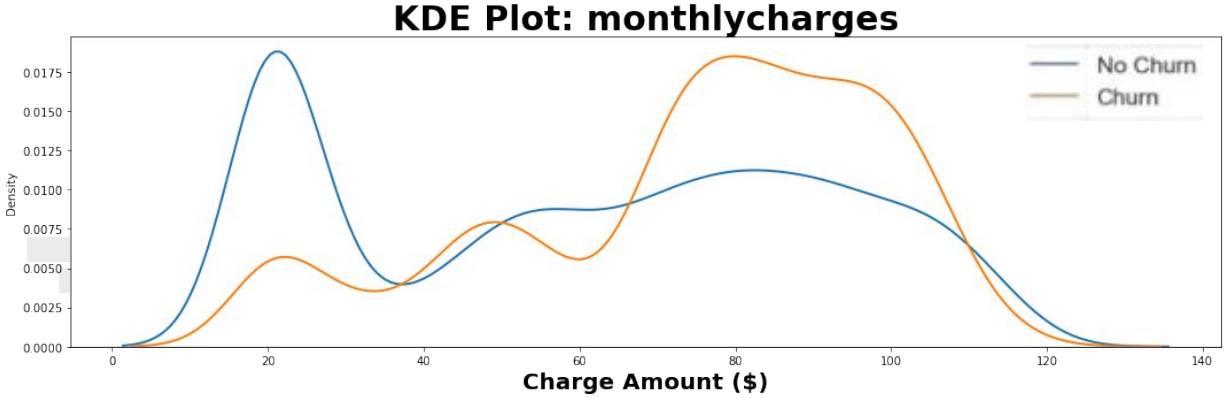
- Customers with streaming services (TV/Movies) churn the most

Customer Tenure and Monthly Charges



- Customers are more likely to churn within the first year of tenure
- As the tenure increases, the probability of churn decreases

- As monthly charges increase, the probability of customer churn increases
- Customers who churn most likely have bills exceeding $60



02

Churn Prediction

Problem 1: Can we predict whether if a customer will churn?

Classification Problem

Since churn is a yes/no variable and not numerical, this is a classification problem.

Classification Models

For classification problems, we will need to train classification models to predict whether a customer will churn.

Additional Data Preparation Steps

OneHotEncoding

In the dataset, we have many categorical values that are Yes/No and they're not suitable for training classification models.

However, just changing Yes/No into 1 or 0 is not good enough as the machine may assume higher numbers are more important. Hence, OneHotEncoding is used.

PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV
No	No phone service	DSL	No	Yes	No	No	No
Yes	No	DSL	Yes	No	Yes	No	No
Yes	No	DSL	Yes	Yes	No	No	No
No	No phone service	DSL	Yes	No	Yes	Yes	No
Yes	No	Fiber optic	No	No	No	No	No

Attempt #1

Standard Train/Test Split

For this attempt, we will be using train_test_split of 30% data reserved for testing.

Parameters Untuned

Parameters are not tuned

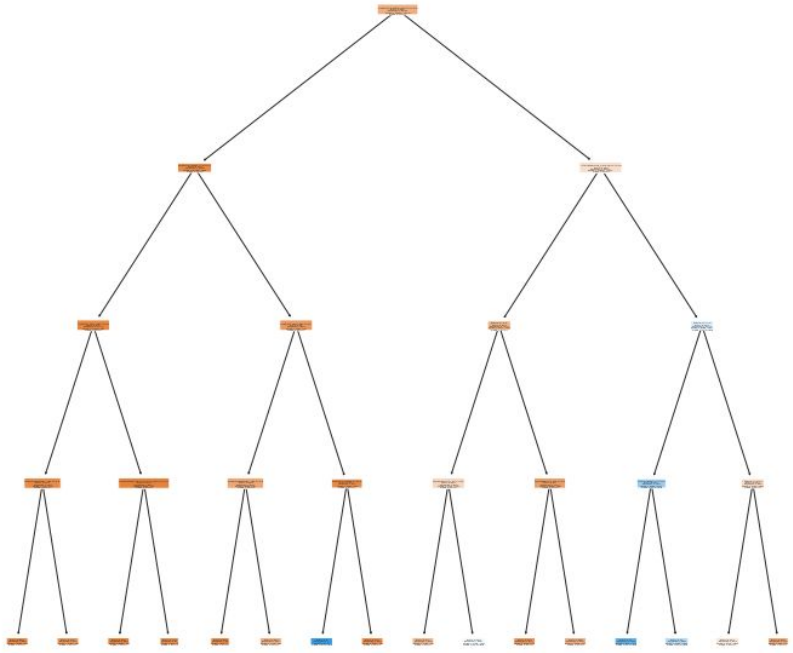
01 **Decision Tree**

02 **Logistic Regression**

03 **Random Forest**

04 **Support Vector Classifier**

01 Decision Tree



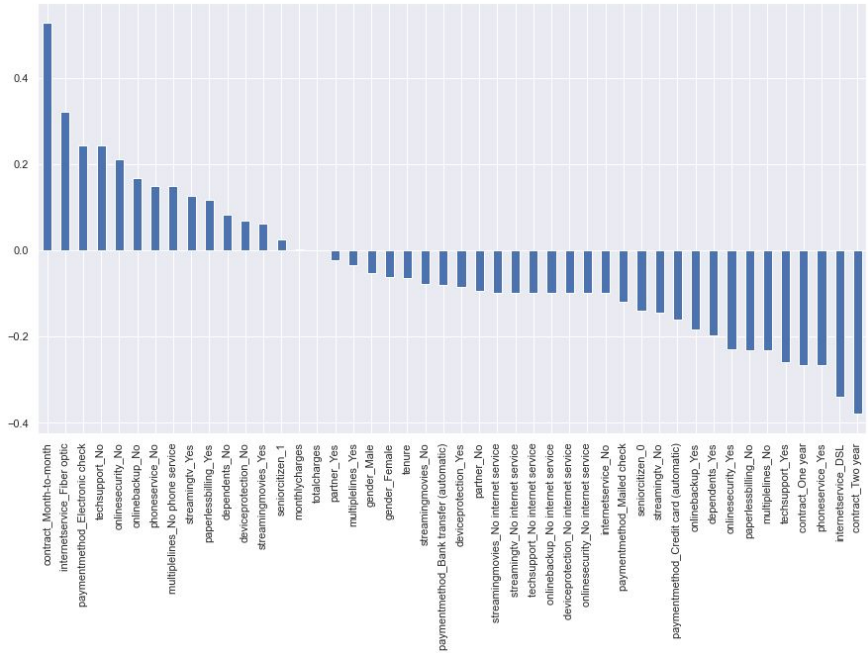
Accuracy measures and rates (Test Data)

Classification Accuracy : 0.7998106956933271

TPR Test : 0.46983546617915906
TNR Test : 0.9150702426564495
FPR Test : 0.08492975734355045
FNR Test : 0.5301645338208409

02

Logistic Regression



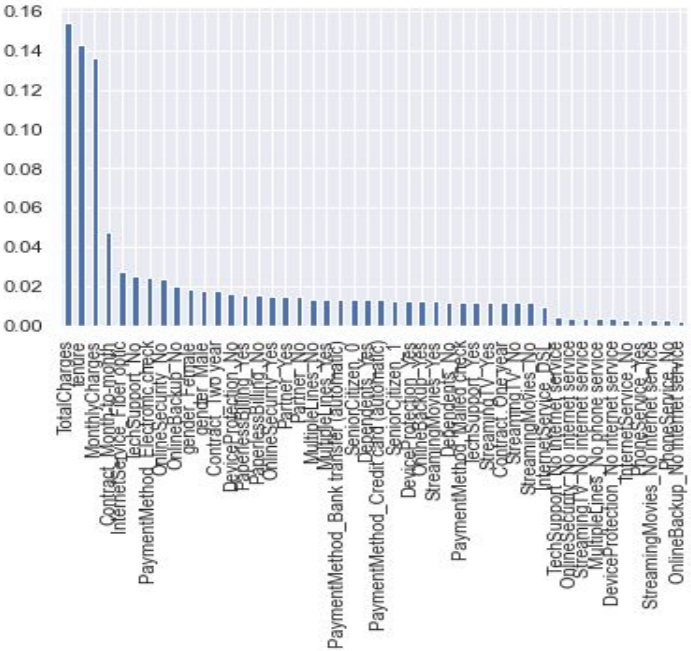
After training: weights of variables for the model

Accuracy measures and rates (Test Data)

Classification Accuracy : 0.8078561287269286

TPR Test : 0.5636042402826855
TNR Test : 0.8972204266321914
FPR Test : 0.10277957336780866
FNR Test : 0.4363957597173145

03 Random Forest



After training: weights of variables for the model

Accuracy measures and rates (Test Data)

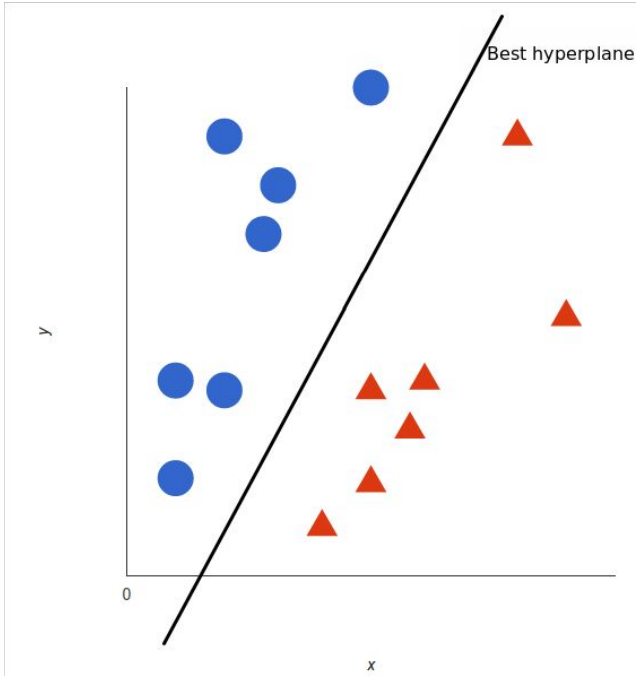
Classification Accuracy : 0.7927117841930904

TPR Test : 0.4835766423357664
TNR Test : 0.9009584664536742
FPR Test : 0.09904153354632587
FNR Test : 0.5164233576642335

04

Support Vector Classifier

How support vector classifier works:



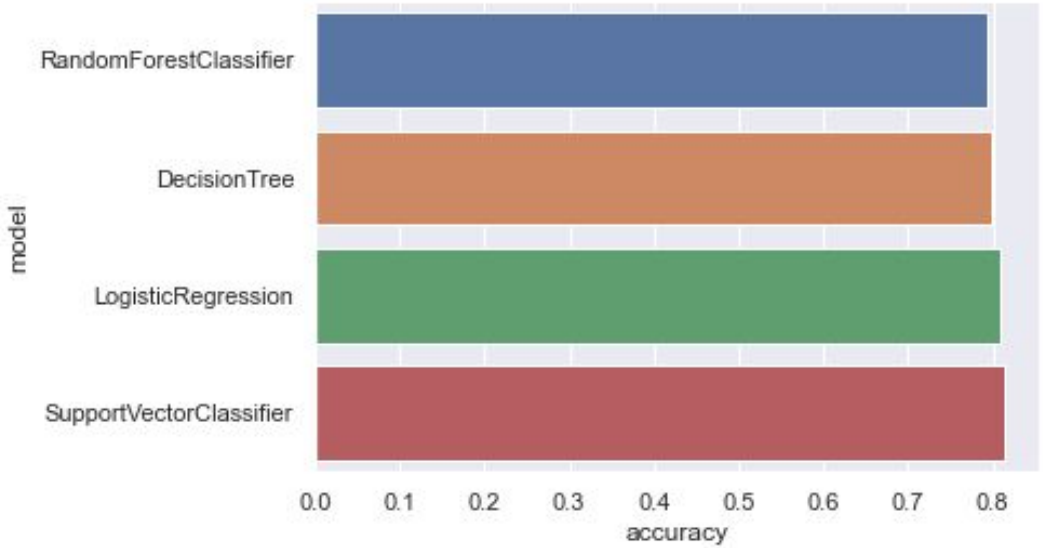
Accuracy measures and rates (Test Data)

Classification Accuracy : 0.8135352579271179

TPR Test : 0.5561797752808989
TNR Test : 0.9005699810006333
FPR Test : 0.09943001899936668
FNR Test : 0.4438202247191011

Comparing Models

	model	accuracy	TPR	TNR	FPR	FNR
0	DecisionTree	0.799811	0.469835	0.915070	0.084930	0.530165
1	LogisticRegression	0.807856	0.563604	0.897220	0.102780	0.436396
2	RandomForestClassifier	0.792712	0.483577	0.900958	0.099042	0.516423
3	SupportVectorClassifier	0.813535	0.556180	0.900570	0.099430	0.443820



High accuracy scores across the board. (shows that the variables in dataset are quite helpful in training the models to predict churn)

However, difficult to pick the best models as different runs results in different best models.

Need more training to get more consistent data (attempt #2)

Attempt #1

Parameters Untuned

Parameters are not tuned

Standard Train/Test Split

For this attempt, we will be using train_test_split of 30% data reserved for testing.

Attempt #2

GridSearchCV

For this attempt, we will find use GridSearchCV to find the best hyperparameters for each model.

K-Folds Testing

Using the most optimised parameter, we will run a k-fold of 10 to compare the models.

Much more accurate & consistent than train_test_split.

Attempt 2:Tuning Models (GridSearchCV)

GridsearchCV will test different depths of the trees to find the best depth with the highest accuracy.

```
# Import GridSearch for hyperparameter tuning using Cross-Validation (CV)
from sklearn.model_selection import GridSearchCV

# Define the Hyper-parameter Grid to search on, in case of Random Forest
param_grid = {'n_estimators': np.arange(100,500,50),    # number of trees
              'max_depth': np.arange(2,7)}              # depth of trees

# Create the Hyper-parameter Grid
hpGrid = GridSearchCV(RandomForestClassifier(),    # the model family
                      param_grid,                  # the search grid
                      cv = 5,                      # 5-fold cross-validation
                      scoring = 'accuracy')        # score to evaluate

# Train the models using Cross-Validation
hpGrid.fit(X_train, y_train.Churn.ravel())
```

```
: GridSearchCV(cv=5, estimator=RandomForestClassifier(),
               param_grid={'max_depth': array([2, 3, 4, 5, 6]),
                           'n_estimators': array([100, 150, 200, 250, 300, 350, 400, 450])},
               scoring='accuracy')
```

The best depth returned by model is a max-depth of 6 for random forest.

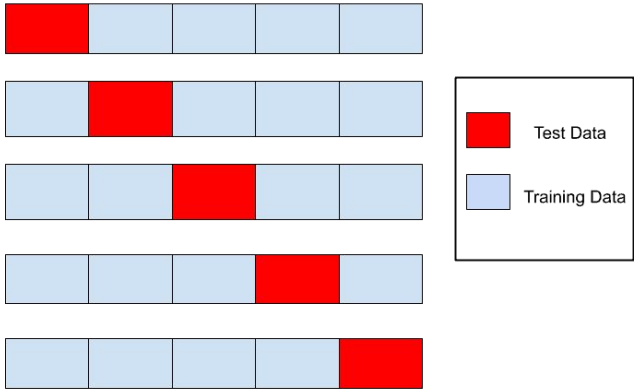
```
|: # Fetch the best Model or the best set of Hyper-parameters
print(hpGrid.best_estimator_)

# Print the score (accuracy) of the best Model after CV
print(np.abs(hpGrid.best_score_))
```

```
RandomForestClassifier(max_depth=6)
0.7997971602434077
```

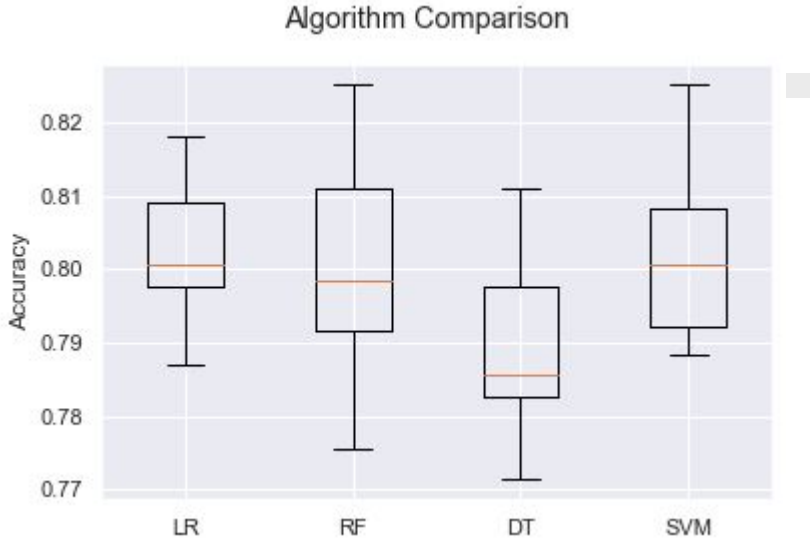
Attempt 2:

K-Folds Cross Validation



Instead of using train test split this time round, a **k-fold cross validation split of 10** is ran for each model.

Boxplot shows the 10 different runs of accuracy score for each model (range of accuracy)



LR - Logistic Regression, RF - Random Forest, DT - Decision Tree, SVM - Support Vector Model/Classifier

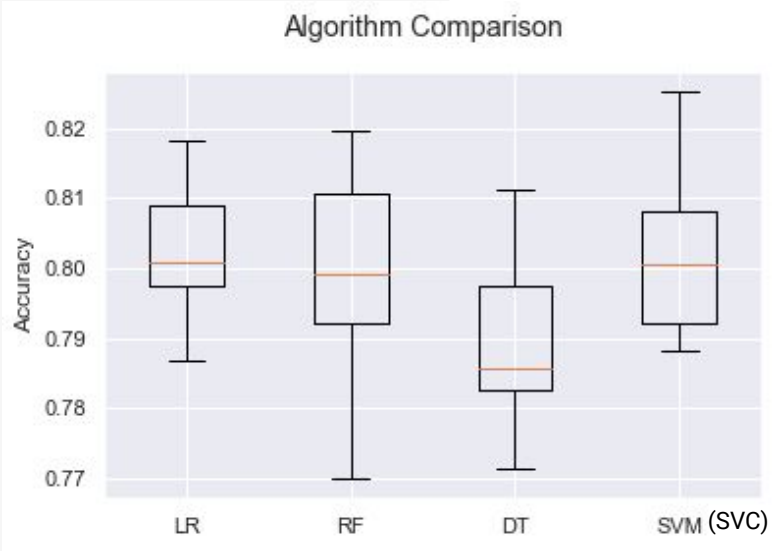
Attempt #2 Results

1 GridSearchCV + Cross Validation Results

Models are still quite close in accuracy, but we can now see which are more consistent with less variation. The two best models are Logistic Regression and Support Vector Classifier(highest median of accuracy and least variation)

2 Predict with Accuracy

Telco can predict future customer data with about ~80% accuracy. If a customer is predicted to churn, they can try to retain them in advance before they churn. (retention > acquisition)



LR - Logistic Regression, RF - Random Forest,
DT - Decision Tree, SVM - Support Vector Model/Classifier

03

Customer Segmentation

Additional Data Preparation Steps

1 Numerical Data

Numerical variables are put on the same scale (sklearn standard scaler) as they have measurements of different units.

This ensures all features are equally considered for the final clustering (i.e. distance measured between points of each feature are roughly the same).

2 Churn variable

Dropped from the dataset to keep the dataset as unlabelled for clustering (unsupervised learning)

Churn
No
No
Yes
No
Yes

c

Problem 2: Can we segment Telco company's customers into clusters in order to achieve more effective customer marketing?

Our dataset and model used

1. Numerical Data
2. Categorical Data

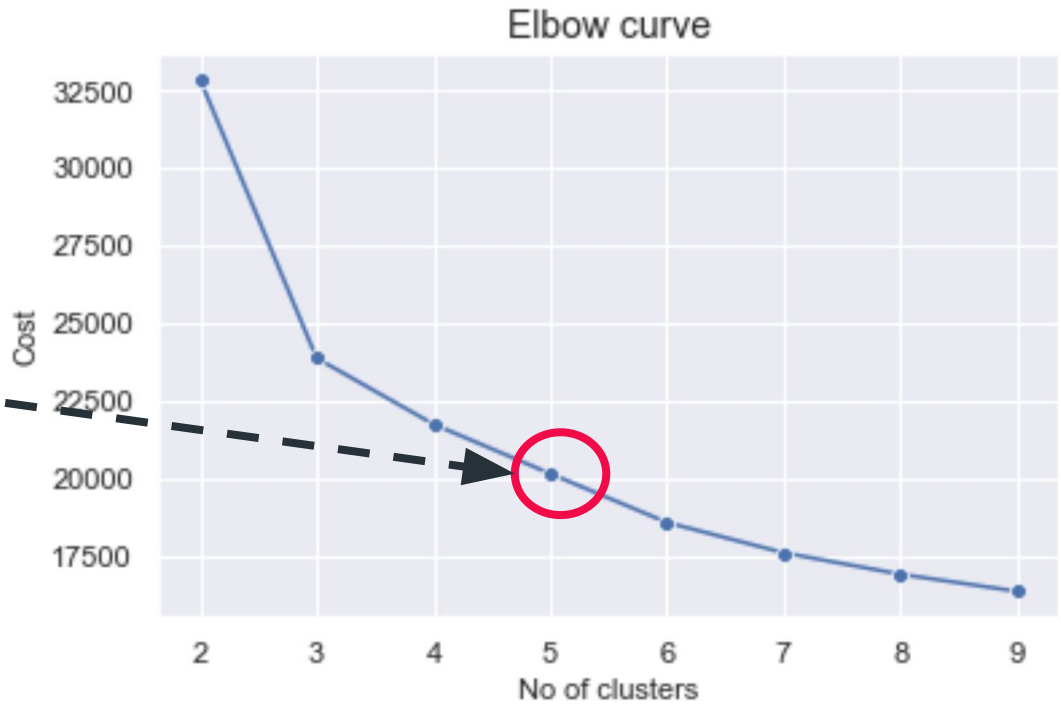
K-Prototype Clustering Method

Finding Optimal Number of Clusters

1 Elbow Method

Elbow point is the cluster number where the value of the cost is not rapidly changing anymore.

Elbow Method → 5 clusters



Finding Optimal Number of Clusters

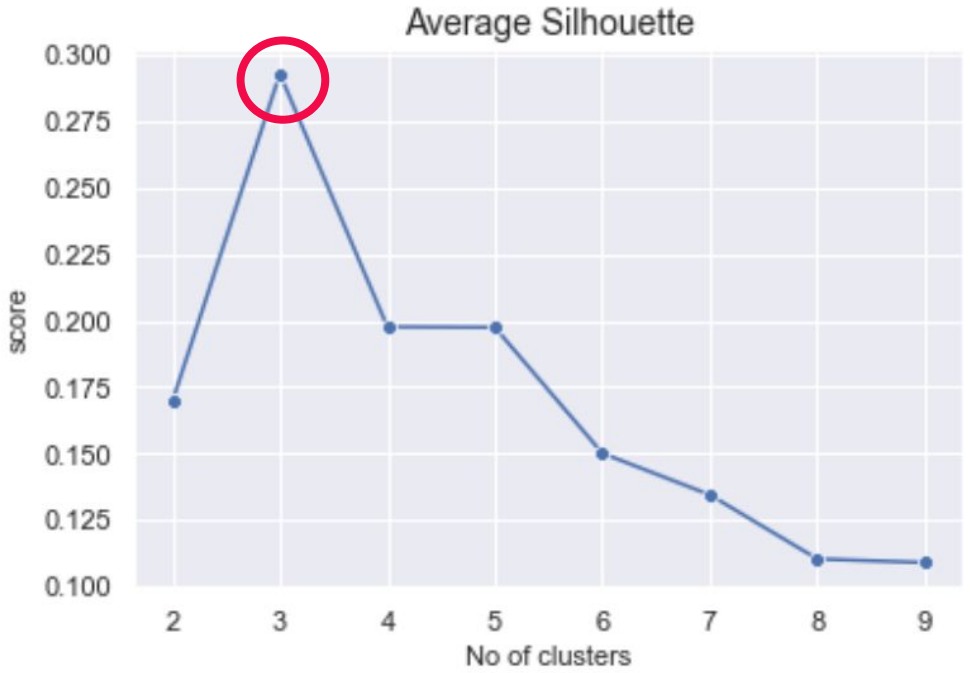
2 Average Silhouette method

A silhouette value close to 1 indicates the data point is well clustered. (i.e. far away from other clustered points).

Average Silhouette method → 3 clusters

Elbow Method → 5 clusters

∴ **We choose 4 clusters as the optimal number of clusters**



K-Prototype Clustering Method

```
kproto = KPrototypes(n_clusters= 4, init='Huang', n_init = 25, random_state=42)
kproto.fit_predict(dfMatrix, categorical= catColumnsPos)
```

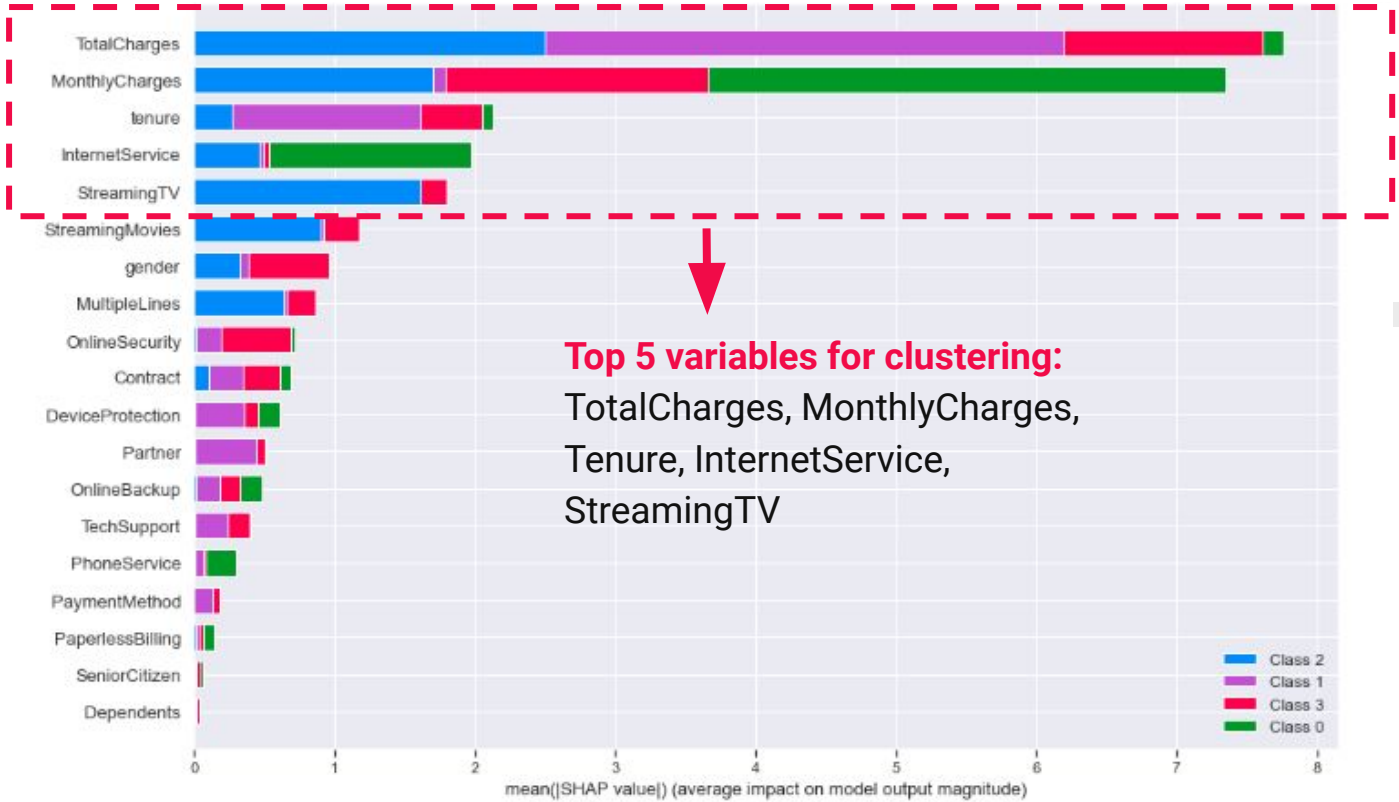
Our results

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Internet
0	Female	0	Yes	No	1	No	No phone service	
1	Male	0	No	No	34	Yes	No	
2	Male	0	No	No	2	Yes	No	
3	Male	0	No	No	45	No	No phone service	
4	Female	0	No	No	2	Yes	No	Fit
...	...	...	...	...	...	...	...	
7038	Male	0	Yes	Yes	24	Yes	Yes	
7039	Female	0	Yes	Yes	72	Yes	Yes	Fit
7040	Female	0	Yes	Yes	11	No	No phone service	
7041	Male	1	Yes	No	4	Yes	Yes	Fit
7042	Male	0	No	No	66	Yes	No	Fit

7043 rows × 21 columns

Machine Learning: Shapley

The Shapley plot ranks the variables in order of importance and it helps us figure out the top 5 features in forming the various clusters.



Top 5 variables for clustering:
TotalCharges, MonthlyCharges, Tenure, InternetService, StreamingTV

Cluster Interpretation for Data Driven Insights

ClusterSegment	Count	TotalCharges	MonthlyCharges	tenure	InternetService	StreamingTV
One	1558	693.819127	21.435366	31.084082	No	No internet service
Two	1867	5530.620648	89.893760	61.563471	Fiber optic	Yes
Three	2067	707.267271	57.607716	13.089985	DSL	No
Four	1551	2055.195551	87.565119	24.219858	Fiber optic	Yes

Cluster 1

- Lowest total charges
- Lowest monthly charges
- Did not subscribe to internet service

Lowest Spenders, Only Subscribe to Phone Services

Cluster 2

- Highest total charges
- Highest monthly charges
- Subscribe to Internet service & StreamingTV

High Spenders, Subscribes to Most Services (High Average Tenure)

Cluster 3

- Largest cluster
- Lowest tenure
- Did not subscribe to any add on services

Average Spenders, Subscribes to Main Services (internet & phone only)

Cluster 4

- High total charges
- High monthly charges
- Low tenure
- Subscribe to Internet service & StreamingTV

High Spenders, Subscribes to Most Services (Low Average Tenure)

Recommendations

Cluster 1 - Lowest Spenders, Only Subscribe to Phone Services

Promote internet services to them or offer bundles consisting of both internet and phone services bundles to these customers

Cluster 2 - High Spenders, Subscribes to Most Services (High Average Tenure)

Offer special services and discounts to these customers to show appreciation for their loyalty.

Cluster 3 - Average Spenders, Subscribes to Main Services (internet & phone only)

Telco can offer more promotions on their add on services to these customers.

Cluster 4 - High Spenders, Subscribes to Most Services (Low Average Tenure)

Introduce Loyalty Programs to ensure customers come back and renew their contracts.

Project Outcome (Summary)

Relationship between variables

The variables: churn, monthly charges, contract type and certain add ons services has a direct impact on whether a customer will churn or not.

Churn Prediction

Given the customer variables, telco can use our classification model (SVC and Logistic Regression) to predict whether a customer will churn or not with ~80% accuracy.

Customer Segments

Telco can use the unsupervised clustering model we provide to better understand their customer segments and execute targeted strategies and recommendations.

THANK YOU!

Team Members and Contributions

Soh Zu Wei - Classification Models (with gridsearchcv, kfold & one-hot-encoding, github, edit video)

Sanskkriti - Problem Motivation, Data Preparation, Cleaning and Exploratory Data Analysis

Jue Lin - Clustering Models (Kprototypes, elbow method, silhouette, clustering strategies)