Predicting Telecom Churn

By : Vandana Jain

Database Used : TELECOM CHURN DATABASE

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No

Data Exploration :

Shape (7043,21)

No of Columns : 21

No of rows   : 7043

Data Cleaning :

dataset.isnull().sum()

customerID	0	gender	0
OnlineSecurity	0	SeniorCitizen	0
Partner	0		
Dependents	0		
OnlineBackup	0		
DeviceProtection	0		

Data which i took was clean.

Converting some categorical columns to numeric values using label encoder

```
# Import label encoder
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
dataset['MultipleLines']= label_encoder.fit_transform(dataset['MultipleLines'])
dataset['MultipleLines'].unique()

dataset['InternetService']= label_encoder.fit_transform(dataset['InternetService'])
dataset['InternetService'].unique()

dataset['gender']= label_encoder.fit_transform(dataset['gender'])
dataset['gender'].unique()

dataset['Partner']= label_encoder.fit_transform(dataset['Partner'])
dataset['Partner'].unique()

dataset['Dependents']= label_encoder.fit_transform(dataset['Dependents'])
dataset['Dependents'].unique()

dataset['StreamingMovies']= label_encoder.fit_transform(dataset['StreamingMovies'])
dataset['StreamingMovies'].unique()

dataset['Churn']= label_encoder.fit_transform(dataset['Churn'])
dataset['Churn'].unique()
```

Model Creation :

Data Split :

Train data : 75%

Test data : 25%

Finding best hyper parameter value using grid_search_KNN.best_params_ For both KNN & Random forest:

KNN

0.7614507181560162 {'n_neighbors': 3}

0.7731879461598005 {'n_neighbors': 5}

0.7868196510994524 {'n_neighbors': 10}

0.7875774060376709 {'n_neighbors': 15}

Winner = {'n_neighbors': 15}

Random Forest

0.7593666681001119 {'n_estimators': 4}

0.7516029500301025 {'n_estimators': 5}

0.7544451133855107 {'n_estimators': 10}

0.7567174822969525 {'n_estimators': 20}

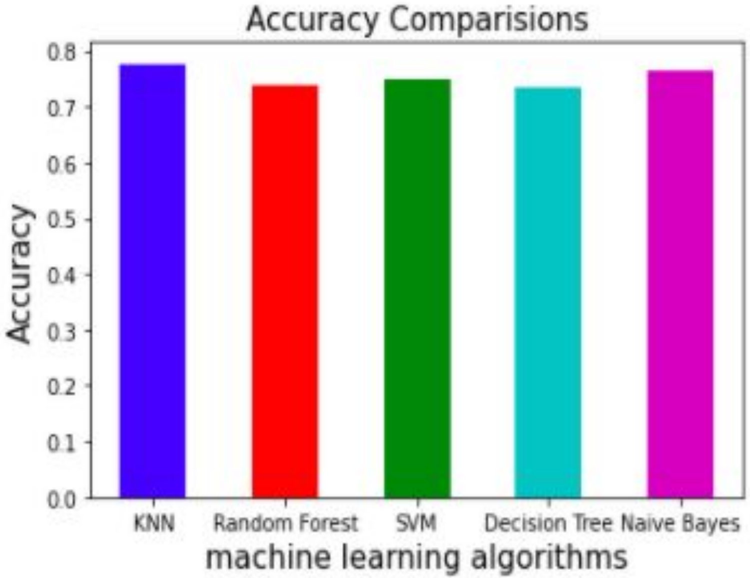
0.7563383360568791 {'n_estimators': 50}

Winner = {'n_estimators': 4}

Model Evaluation :

Bar Graph between the accuracy of algorithms

When took 7 column values :

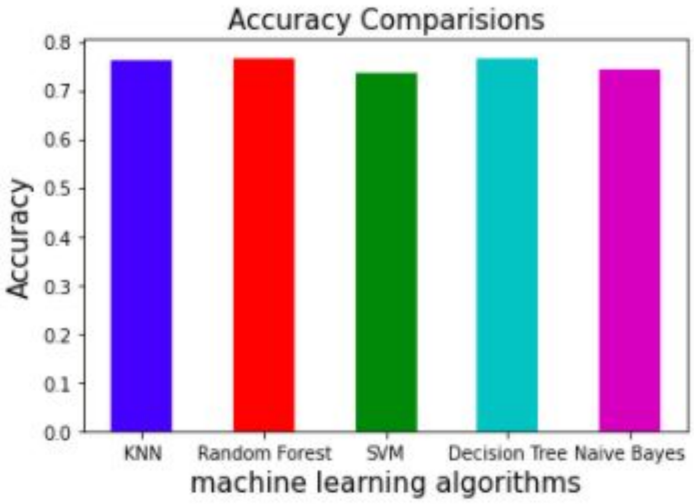


Winner here is :

KNN

Bar Graph between the accuracy of algorithms

When took 2 column values :



Winner here is :
Decision Tree

Note :- Many features have a strong correlation with the 'Churn' variable. For example, the customers that have a 'Month to Month' contract are more likely to churn as it gives customers a lot of flexibility and allows them to leave the given operator at any time.

So I decided to take “monthly charges “column & “Tenure “column to see some improvement in accuracy of the model & I found the improvement but surprisingly very little.

Comparison of algorithms :

Algorithms	Accuracy When Column taken=7	Accuracy When Column taken=2
KNN	**0.7677455990914254** (winner)	0.7626348665530949
Random Forest	0.7410562180579217	0.7632027257240205
SVM	0.7490062464508802	0.7370812038614424
Decision Tree	0.7365133446905168	0.7660420215786485
Naive Bayes	0.7626348665530949	0.7444633730834753

Thank You