Coursera Advanced Data Science Capstone

MOHAMMED JABER EL FILALI

USE CASE

- Telecommunication Companies usually want to understand the behavior of their customers and how they respond to their pricing a new services (Positively – Negatively), and develop Costumer Retention Programs
- Goal : Predict whether a customer will churn or not based on his subscriptions to services, payments, and other features ..
- NB : This project is experimental and no deployment is necessary

Data Source

- The Database if from Kaggle, known as Telco Customer Churn : https://www.kaggle.com/blastchar/telco-customer-churn
- Contains 7043 Rows and 21 Features:
- The Churn Column “Yes or No” is the Target
- The data set includes information about:
  - Customers who left within the last month – the column is called Churn
  - Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
  - Customer account information – how long they’ve been a customer, contract, payment method, paperless billing, monthly charges, and total charges
  - Demographic info about customers – gender, age range, and if they have partners and dependents

Solution

- Classification Model trained to predict whether a Customer leaves or stays the next month
- Each step was made as a function and saved in the server
- The Model is trained and saved in the memory of the Entreprise server for later usage

Solution - Notebook

- The needed dependencies and the data are imported

```
import types
import pandas as pd
from ibm_botocore.client import Config
import ibm_boto3
import numpy as np
from sklearn.preprocessing import StandardScaler
import dill
from joblib import load

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share your notebook.
client_e593f080a7a74a92a3fbe74ef2ba4af0 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='d8MCA5cpOB1LbznjKG5Pzr1JKA1ojKb46zFKnUC4VDrH',
    ibm_auth_endpoint="https://iam.bluemix.net/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_e593f080a7a74a92a3fbe74ef2ba4af0.get_object(Bucket='churnproject-donotdelete-pr-nwwz91i
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
```

Solution - Notebook

- The preprocessing and modelling steps are uploaded from their persisted files

```
#EDA
file = open('dataexp_function.bin', 'rb')
dataexp = dill.load(file)
file.close()

#Feature Engineering 1
file = open('featеng_1_function.bin', 'rb')
feat_eng1 = dill.load(file)
file.close()

#Featuer Engineering 2
file = open('feateng_2_function.bin', 'rb')
feat_eng2 = dill.load(file)
file.close()

#GBOOST Model
model = load('GBoost_bestmodel.joblib')
```

Solution - Notebook

- Then we create a function for preparing and predicting using these persisted functions

```
def preprocessing(data):
    exp_data = dataexp(data)
    feateng_data = feat_eng1(exp_data)
    preprocessed_data = feat_eng2(feateng_data)
    return preprocessed_data


preprocessed_df = preprocessing(df)

def prediction(customer):
    return model.predict(np.array(customer.drop(['customerID','Churn'], axis=1)))
```

Solution - Notebook

- Now we test the model on a customer

```
customer = preprocessed_df[preprocessed_df['customerID']=='7590-VHVEG']
if customer['Churn'].values[0]==0:
    print('The customer is leaving')
else:
    print('The customer is staying')

#Testing the Model
model_prediction = prediction(customer)[0]

if model_prediction==0:
    print('This customer is likely to leave')
else:
    print('This customer is likely to stay')
```

```
The customer is leaving
This customer is likely to leave
```

Architectural Choices Development Environment

Data Persistence	Execution Engine	Host	Coding Envt	Coding Language

Architectural Choices
Data Exploration

NumPy

Numerical data

Pandas

Importing data

Data management

Statistical Moments

matplotlib

Histograms

seaborn

Count plot

Boxplots

Architectural Choices
Feature Engineering

NumPy

- Binning

Pandas

- One hot Encoding

scikit learn

- sklearn.preprocessing:
  Standard Scaler

Architectural Choices
Modelling

-Deep Learning :

To create a CNN

Easy and handy to use

-Machine Learning:

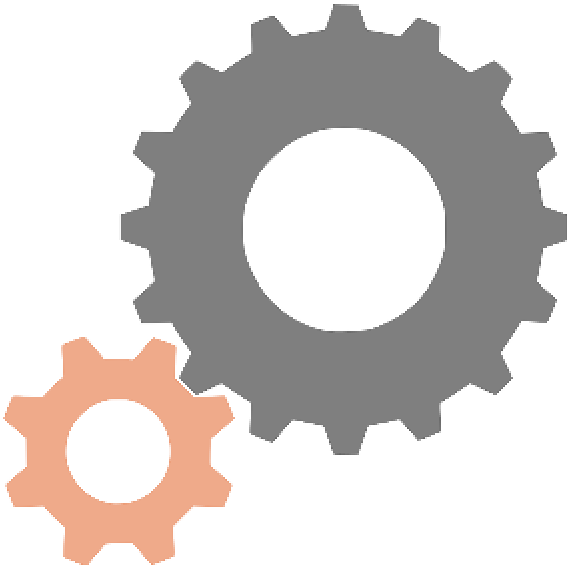
Different classification models

Easy to use (Few LOCs)

Architectural Choices
Persistence

- Dill API :
  - For functions persistence

- Joblib API :
  - For Model Persistance



Exploratory Data Analysis
Categorical Features

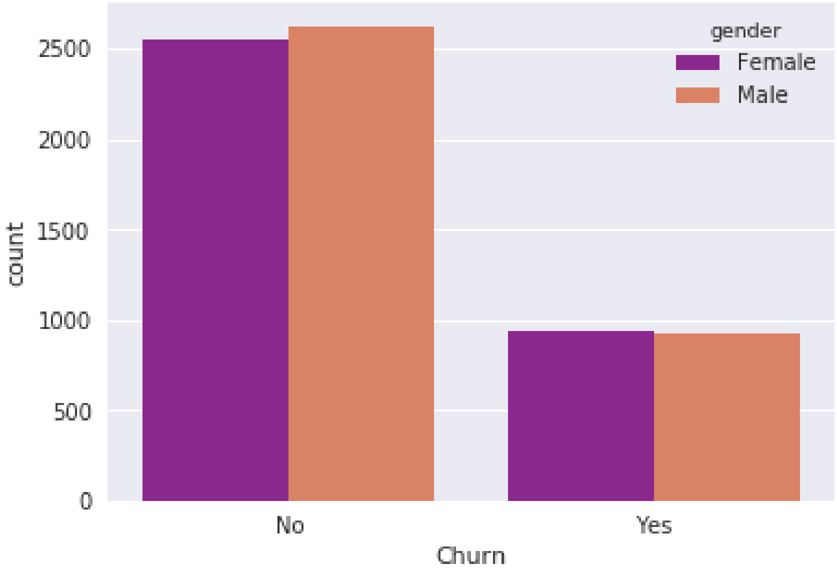
- customerID : Customer’s ID
- Gender : Whether the customer is a male or a female
- SeniorCitizen : Whether the customer is a senior citizen or not (1, 0)
- Partner : Whether the customer has a partner or not (Yes, No)
- Dependents : Whether the customer has dependents or not (Yes, No)
- PhoneService : Whether the customer has a phone service or not (Yes, No)
- MultipleLines : Whether the customer has multiple lines or not (Yes, No, No phone service)
- InternetService : Customer’s internet service provider (DSL, Fiber optic, No)
- OnlineSecurity : Whether the customer has online security or not (Yes, No, No internet service)

Exploratory Data Analysis
Categorical Features

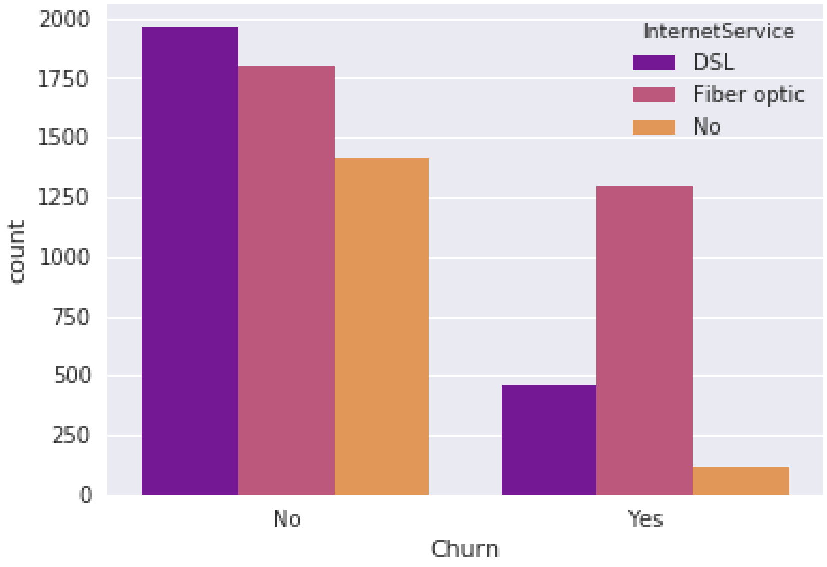
- OnlineSecurity : Whether the customer has online security or not (Yes, No, No internet service)
- OnlineBackup : Whether the customer has online backup or not (Yes, No, No internet service)
- DeviceProtection : Whether the customer has device protection or not (Yes, No, No internet service)
- TechSupport : Whether the customer has tech support or not (Yes, No, No internet service)
- StreamingTV : Whether the customer has streaming TV or not (Yes, No, No internet service)
- StreamingMovies : Whether the customer has streaming movies or not (Yes, No, No internet service)
- Contract : The contract term of the customer (Month-to-month, One year, Two year)
- Paperless : BillingWhether the customer has paperless billing or not (Yes, No)
- PaymentMethod : The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- **Churn** : Whether the customer churned or not (Yes or No) **–Target Feature -**

Exploratory Data Analysis
Categorical Features – Visualizations

- Count plots for all the categorical features to see the power of prediction, for example :



The gender is not a good predictor,



Internet Service is a good predictor

Exploratory Data Analysis
Numerical Features

- Tenure : Number of months the customer has stayed with the company
- MonthlyCharges : The amount charged to the customer monthly
- TotalCharges : The total amount charged to the customer

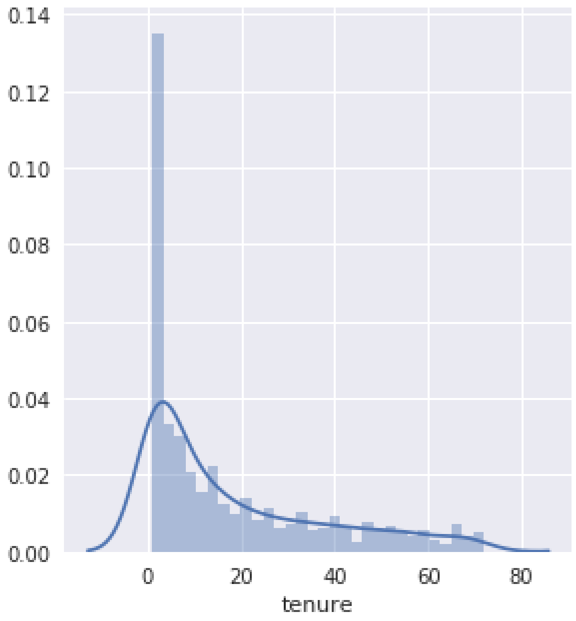
Exploratory Data Analysis Numerical Features – Statistical Moments

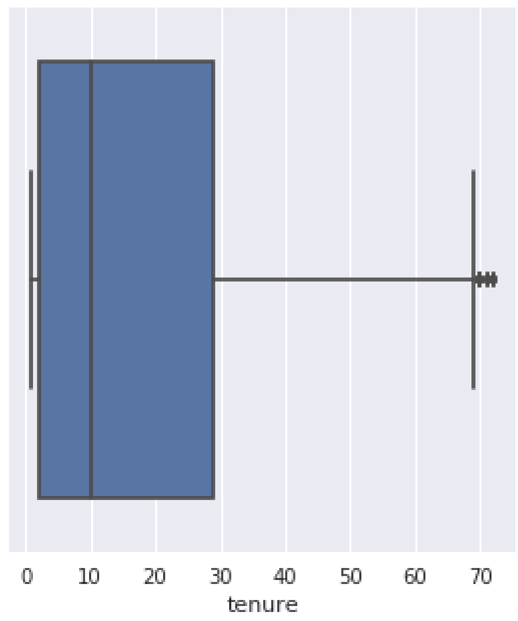
	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

Exploratory Data Analysis
Numerical Features - Visualizations

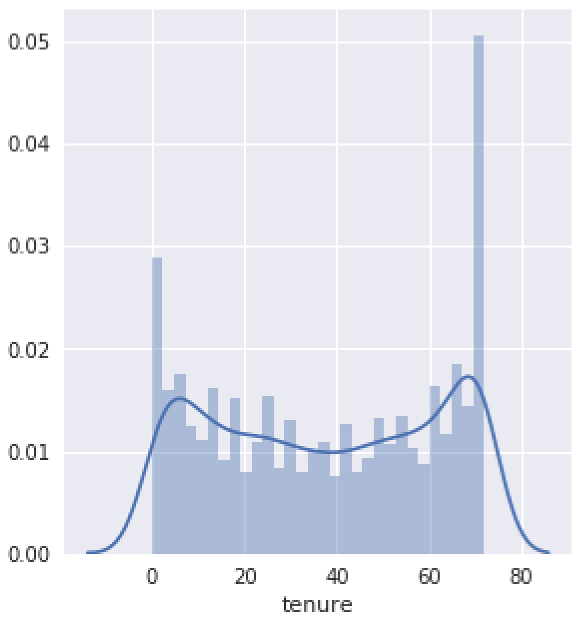
- Histograms, boxplots and heatmap for correlation, for example, the feature Tenure :

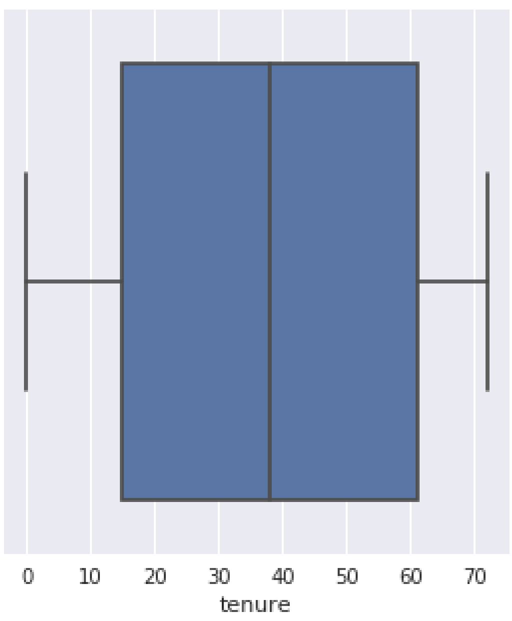
Churn





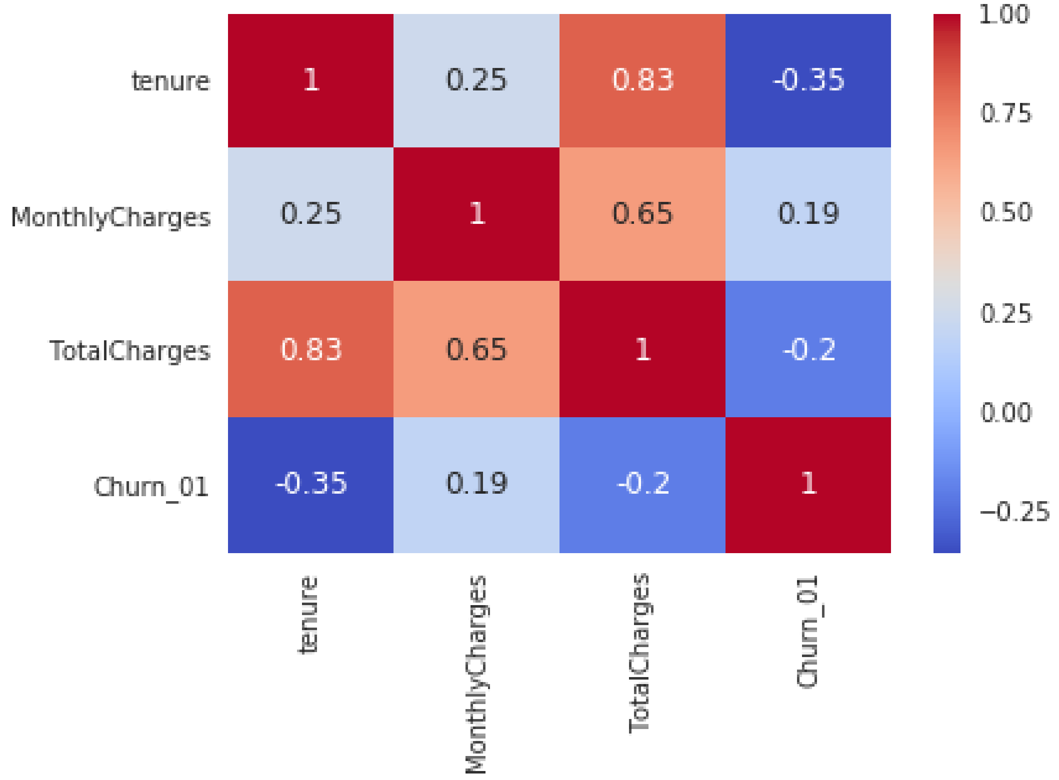
No churn





Exploratory Data Analysis
Numerical Features - Visualizations

- Correlation



Exploratory Data Analysis
Data Quality

- No missing values for the majority of features

- For total charges, the data was string and the missing values were replaced by space → That was easy to solve

Feature engineering

- Feature Creation
  - From MonthlyCharges → YearlyCharges = MonthlyCharges * 12

- One Hot Encoding for Categorical Features

Modelling
Machine Learning

- Used classifiers :
  - Logistic Regression
  - Gradient Boosting
  - Random Forest
  - Naïve Bayes
  - SVM
  - KNN
- Metrics:
  - Accuracy and Precision

	Model	Training Accuracy	Validation Accuracy
0	LogReg	0.806790	0.808677
1	Gboot	0.815677	0.796586
2	RandForest	0.815855	0.797297
3	NaiveBayes	0.700320	0.719061
4	SVM	0.710451	0.698435
5	KNN	0.809101	0.773826

	Model	Training Precision	Validation Precision
0	LogReg	0.669579	0.647260
1	Gboot	0.840228	0.731544
2	RandForest	0.710018	0.642276
3	NaiveBayes	0.468030	0.468254
4	SVM	0.475850	0.439153
5	KNN	0.692644	0.571429

Modelling
Deep Learning

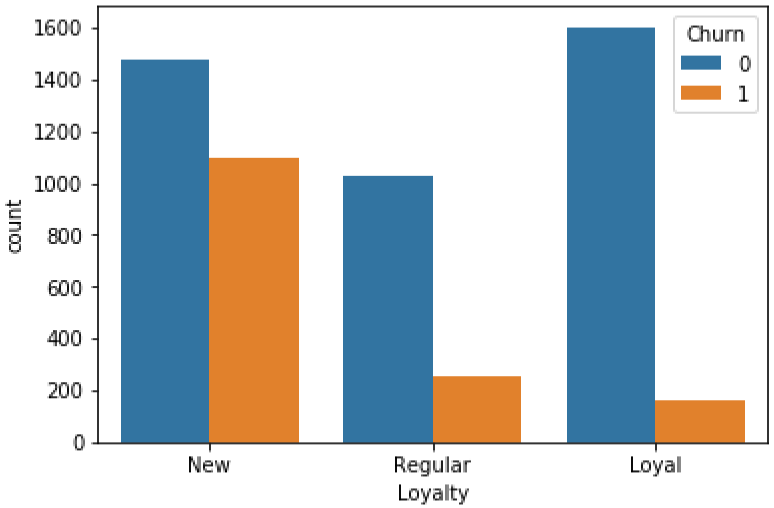
- Used Classifier :
  - Convolutional Neural Net with three layers
- Optimization and Activations functions :
  - Activation functions : sigmoid, tanh, relu
  - Optimizers : RMSProp, Adadelta, Adagrad, Adam, Nesterov Adam
- Result :

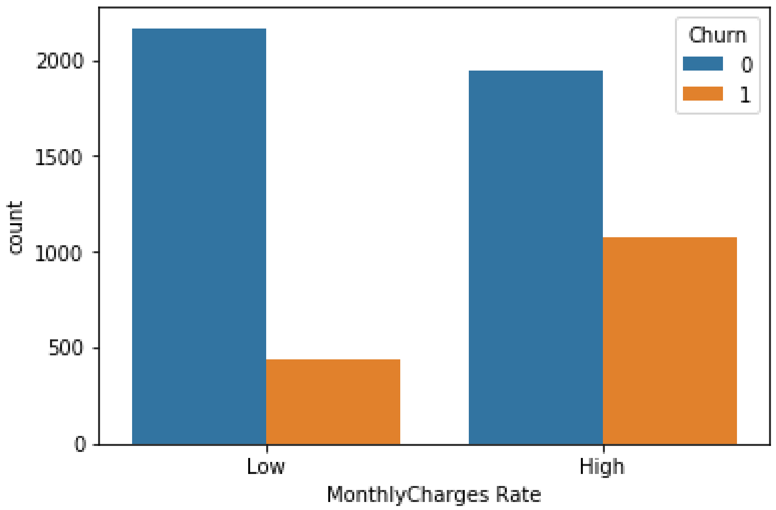
	Act Layer1	Act Layer2	Optimizer	Accuracy
1	sigmoid	sigmoid	adagrad	0.790185

- Conclusion : Work with **Machine Learning** ? Least costly and deep learning doesn't prove to be better

Feature Engineering II

- Feature Creation :
  - Loyalty feature from 'Tenure' : New, Regular, Loyal
  - Monthly Charges : Low, Medium, High





- Scaling using Sklearn's standard scaler

Modelling II
Model improvement ?

	Model	Training Accuracy	Validation Accuracy
0	LogReg	0.806790	0.808677
1	Gboot	0.815677	0.796586
2	RandForest	0.815855	0.797297
3	NaiveBayes	0.700320	0.719061
4	SVM	0.710451	0.698435
5	KNN	0.809101	0.773826

	Model	Training Precision	Validation Precision
0	LogReg	0.669579	0.647260
1	Gboot	0.840228	0.731544
2	RandForest	0.710018	0.642276
3	NaiveBayes	0.468030	0.468254
4	SVM	0.475850	0.439153
5	KNN	0.692644	0.571429

	Model	Training Accuracy	Validation Accuracy
0	LogReg	0.808567	0.804410
1	Gboot	0.798436	0.797297
2	RandForest	0.802702	0.802276
3	NaiveBayes	0.712940	0.733997
4	SVM	0.807145	0.805832
5	KNN	0.803235	0.788762

	Model	Training Precision	Validation Precision
0	LogReg	0.676352	0.639860
1	Gboot	0.820946	0.753623
2	RandForest	0.687384	0.662447
3	NaiveBayes	0.480644	0.484140
4	SVM	0.678601	0.649635
5	KNN	0.648866	0.597315

Model Choice ?

- Gradient Boosting
- Why ?
  - Good in Accuracy and Precision

Model Training and Evaluation

- Hyperparameter Tuning for Gradient Boosting :
  - The learning rate
  - The number of estimators,
  - Maximum depth
  - Minimum sample split
  - Minimum sample leaf
  - Rate of subsample.
- Achieved Precision : **90%**

THANK YOU !