Report by:

Madan Mohan
1NC20IS018
Dept. of ISE
NCET

Abstract

Churn prediction is predicting which customers are at high risk of leaving your company or canceling a subscription to a service, based on their behavior with your product.

Customer churn analysis and prediction in some sector is an issue now a days because it’s very important for industries to analyze behaviors of various customer to predict which customers are about to leave the subscription from their company.

So machine learning techniques and algorithm plays an important role for companies in today’s commercial conditions because gaining a new customer’s cost is more than retaining the existing ones.

This project focuses on machine learning techniques for predicting customer churn through which we can build the classification model such as K-Nearest Neighbor.

Keywords- churn, machine learning, K-Nearest neighbor.

Chapter 1

Introduction

The customer who ceases a product or service for a given period is referred as churner. In a company, the individual who has opted service from a firm is referred to as Churn.

The individual who probably intends to depart from the firm in near future was predicted by the churn model. Many industries build a model like a churn as a common application for data mining technique. Mobile telephone organizations present across the globe are almost on the verge of building their own churn model. Furthermore, to retain the customers, churn results can be efficiently utilized for various other goals.

Churn Management approach is actually the first step in building a model. In general, the project needs a churn model in the best way instead of taking a single method which has the best lift. So here we have built an automated application as a default for a long run.

In this digital era, the client of one company may also be a consumer of one or more telecommunication firms. Some of us may use different carriers based on the distance and some others may use different carriers based on the different plans they offer. While performing the analysis using machine learning customer experience tends to provide valuable insights. Some people will change their service providers from time to time. Increase or decrease in the calling rate will also depend on different job responsibilities. Based on the availability of the data various situations may reflect.

Chapter 2

Tools Exposed

2.1 Jupyter notebook

The juypter notebook app is a server-client application that allows editing and running notebook documents via a web browser. The jupyter notebook app can be executed on a local desktop requiring no internet access or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the jupyter notebook app has a dashboard, a control panel showing local files and allowing to open notebook documents or shutting down their kernels.

A notebook kernel is a computational engine that executes the code contained in a notebook document. The jupyter kernel referenced in this guide executes python code. Kernels for many other language exist. When you open a notebook document the associated kernel is automatically launched. When the notebook is executed the kernel performs the computation and produces the results.

The jupyter notebook extends the console based approach to interactive computing in a qualitatively new direction, providing a web based application suitable for capturing the whole computation process: developing, computing and executing code as well as communicating the results. The jupyter notebook combines two components a web application and notebook documents.

A web application: A web browser based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output. Notebook documents: A representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images and rich media representation of objects.

2.2 Google colab

Colaboratory or colab for short, is a product from Google research. Colab allows anybody to write and execute arbitrary python code through the browser and is especially well suited to machine learning, data analysis and education. More technically colab is a hosted jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

Colab resources are not guaranteed and not unlimited, and the usage limits sometimes fluctuate. This is necessary for colab to be able to provide resources free charge. Resources in colab are prioritized for interactive use cases. We prohibit actions associated with bulk compute, actions that negatively impact others as well as actions associated with bypassing the policies. Jupyter is the open source project on which the colab is based. Colab allows you to use and share jupyter notebooks with others without having to download, install or run anything.

You can search colab notes using google drive. Clicking on the colab logo at the top left of the notebook view will show all notebooks in drive. You can also search for notebooks that you have opened recently by clicking on file and then open notebook. Google drive operations can time out when the number of folders or subfolders in a folder grows too large. If thousands of items are directly contained in the top level “My drive” folder then mounting the drive will likely time out. Repeated attempts may eventually succeed as failed attempts cache partial state locally before timing out. Colab is able to provide resources free of cost in part by having dynamic usage limits that sometimes fluctuate this means that overall usage limits as well as idle timeout periods, maximum VM lifetime, GPU types available and other factors vary over time. Colab does not publish these limits in parts because they can vary quickly. This is necessary for colab to be able to provide access these resources free of charge. Colab works with most of the major browsers and is most thoroughly tested with the latest versions of Chrome, Firefox and Safari.

Chapter 3

Task Performed: Data Analysis of Churn Prediction

3.1 General Steps

- Importing Libraries and Loading the dataset.
- Check for the NULL values and perform data cleaning.
- Convert categorical data into numerical data.
- Check features Co-relation.
- Extract features (x) and labels (y) from data set.
- Split the data into train and test, then do feature scaling.
- Train the model by using K-Nearest Neighbor model.
- Check Accuracy and Outputs.

3.2 Importance of data analysis

While analyzing data sets, it is important to define the objectives so that further steps become clearer. Analysis lets us pose questions about data. For questioning data, it is important to have data collection on which further operations will be carried out. After the above steps, “Data Analysis comes into picture. Data analysis is the process of raw data cleaning and conversion so that further operations become easier to carry on and then the conclusions can be drawn from the results. For Today, data has become the backbone of all research in almost every field. Research and analysis is no more limited to just the area of sciences, but has grown to be a part of businesses – startups and established organizations, government works and more.

Data set:

A data set is a collection of similar and related data or information. It is organized for better accessibility of an entity. Data sets are used for data analytics as they provide related information in a united form. It can be structured or unstructured.

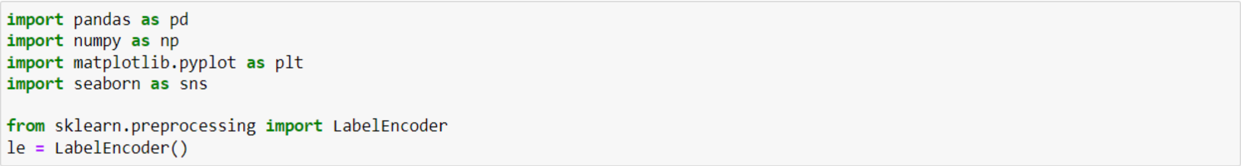
Data set link: https://drive.google.com/drive/folders/1-uFf-k5jh__Po3YRncPBu3uejp1yM-SM

File name: Churn_Modelling.csv

Chapter 4

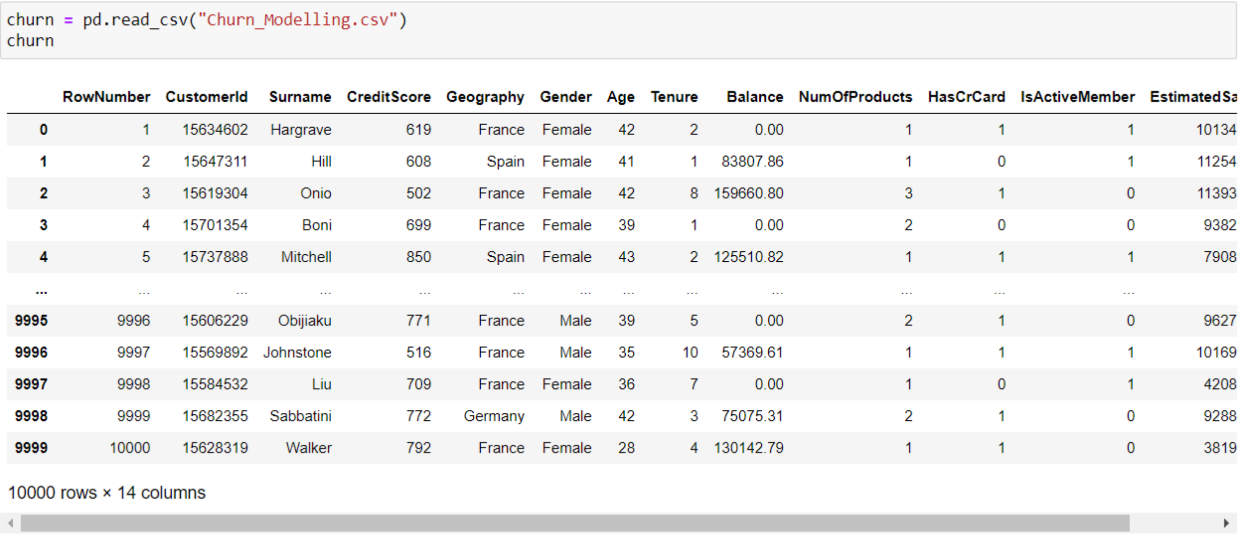
Results and Discussions

4.1 Import the libraries



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

4.2 Loading the Data Set

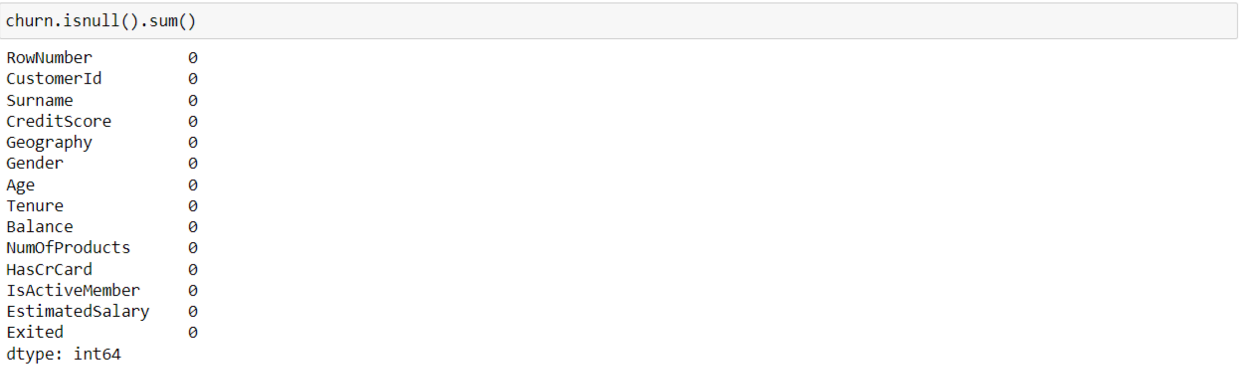


```
churn = pd.read_csv("Churn_Modelling.csv")
churn
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSa
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	10134
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	11254
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	11393
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	9382
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	7908
...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	9627
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	10169
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	4208
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	9288
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	3819

10000 rows × 14 columns

4.3 Check for NULL values and perform Data Cleaning

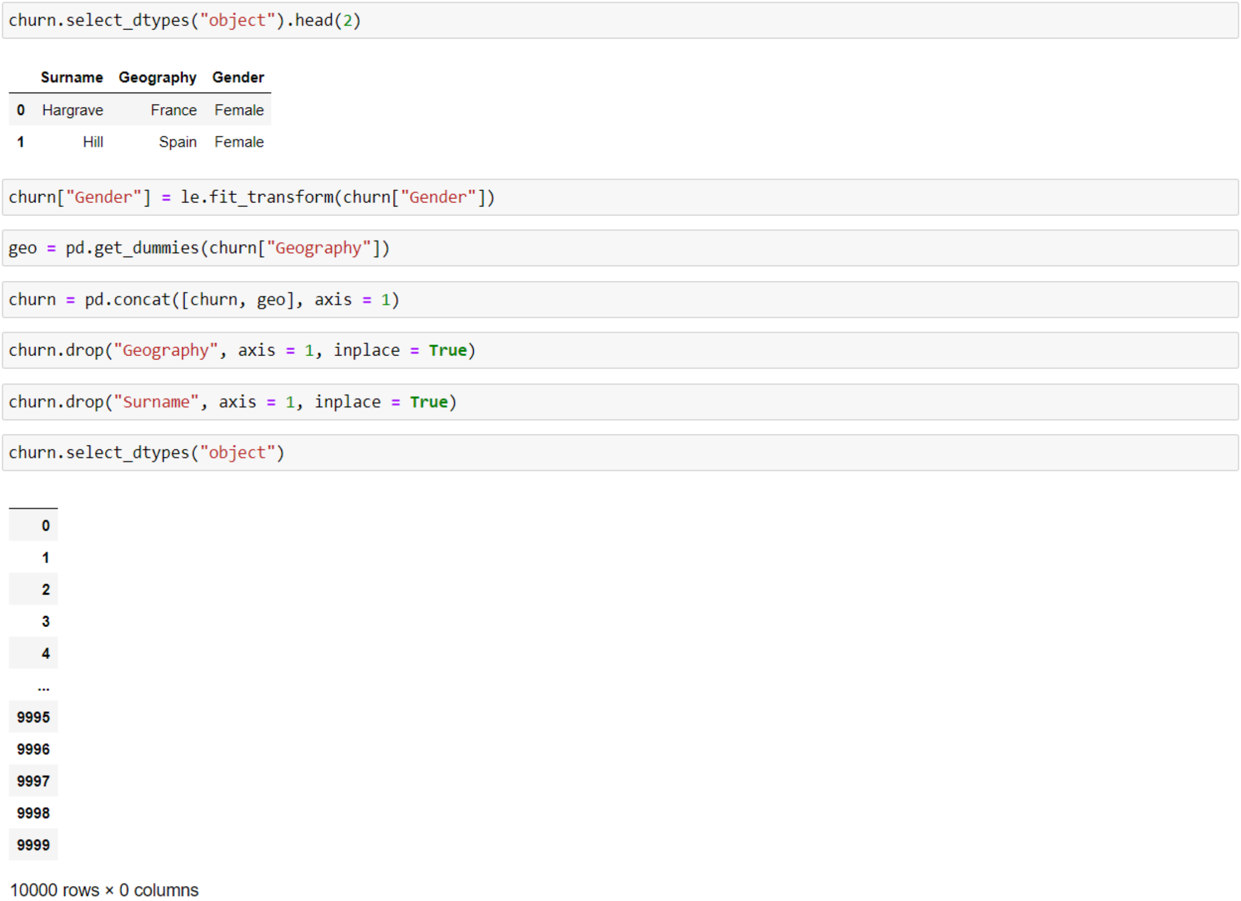


```
churn.isnull().sum()
```

```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

4.4 Convert the Categorical Data into Numerical Data

Here we convert the Categorical Data into Numerical data and we remove the Attributes which are not fit for the model building.



```
churn.select_dtypes("object").head(2)
```

	Surname	Geography	Gender
0	Hargrave	France	Female
1	Hill	Spain	Female

```
churn["Gender"] = le.fit_transform(churn["Gender"])
```

```
geo = pd.get_dummies(churn["Geography"])
```

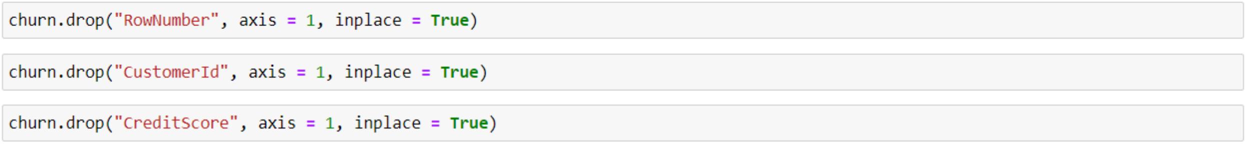
```
churn = pd.concat([churn, geo], axis = 1)
```

```
churn.drop("Geography", axis = 1, inplace = True)
```

```
churn.drop("Surname", axis = 1, inplace = True)
```

```
churn.select_dtypes("object")
```

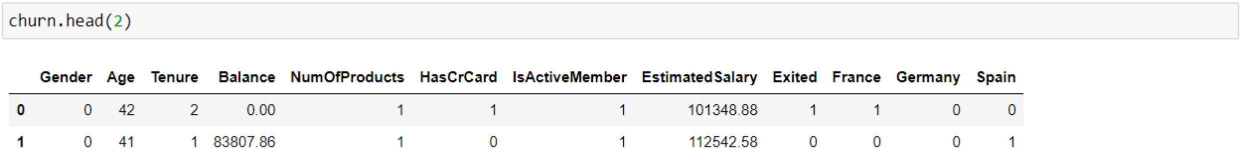
```
   0
   1
   2
   3
   4
 ...
9995
9996
9997
9998
9999

10000 rows × 0 columns
```



```
churn.drop("RowNumber", axis = 1, inplace = True)
```

```
churn.drop("CustomerId", axis = 1, inplace = True)
```

```
churn.drop("CreditScore", axis = 1, inplace = True)
```

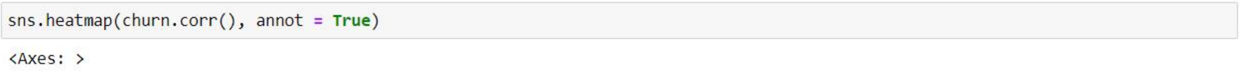


```
churn.head(2)
```

	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	France	Germany	Spain
0	0	42	2	0.00	1	1	1	101348.88	1	1	0	0
1	0	41	1	83807.86	1	0	1	112542.58	0	0	0	1

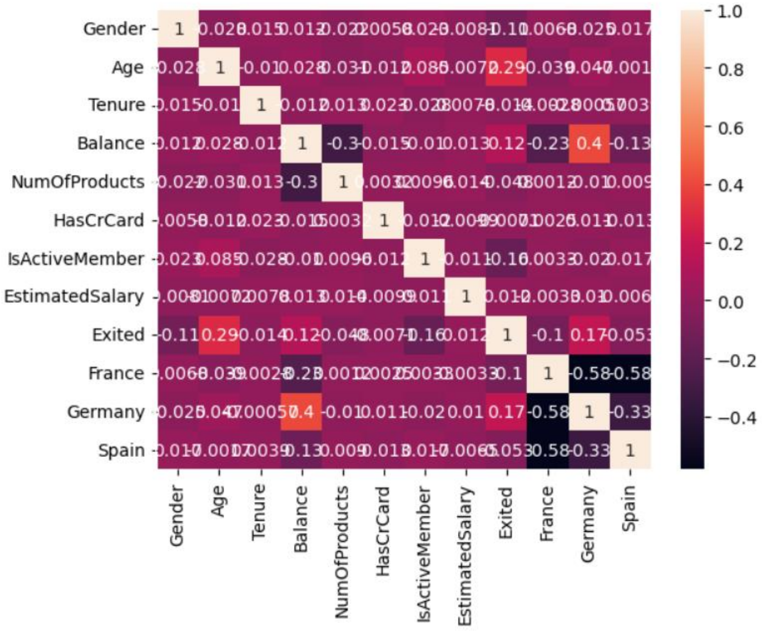
Now there is no Categorical data in the given data set.

4.5 Check Features Co-relation



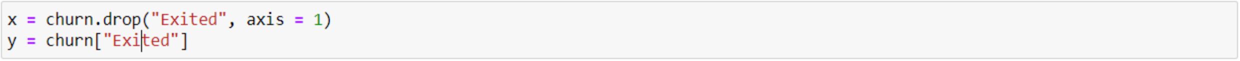
```
sns.heatmap(churn.corr(), annot = True)
```

```
<Axes: >
```



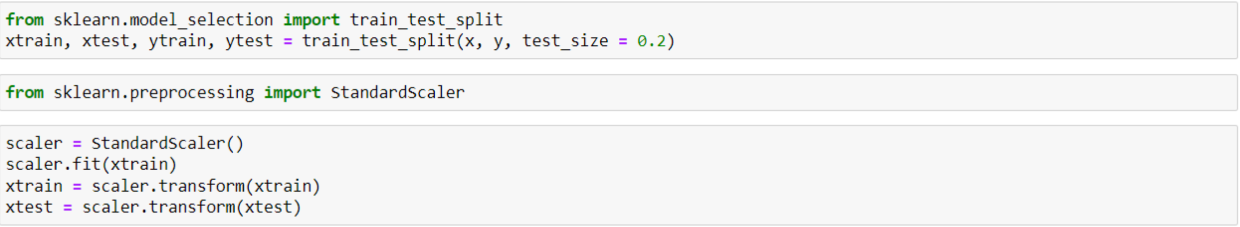
4.6 Extract Features (x) and Labels (y) from data set

Here we train the data to find the EXITED ones in the data set.



```
x = churn.drop("Exited", axis = 1)
y = churn["Exited"]
```

4.7 Split the data into train and test, then do feature scaling

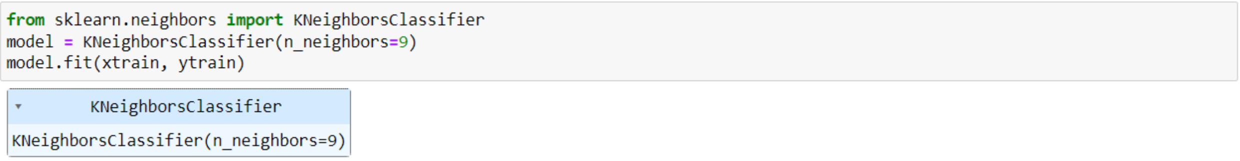


```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.2)
```

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
scaler.fit(xtrain)
xtrain = scaler.transform(xtrain)
xtest = scaler.transform(xtest)
```

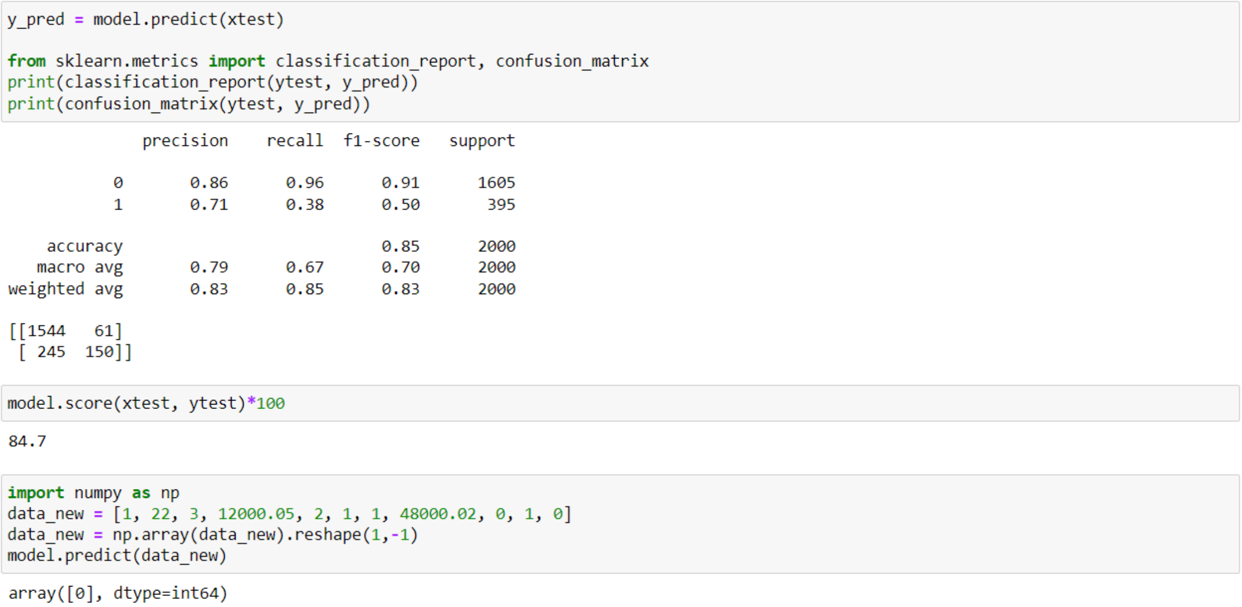
4.8 Train the model by using K-Nearest Neighbor model



```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=9)
model.fit(xtrain, ytrain)
```

```
▾        KNeighborsClassifier
KNeighborsClassifier(n_neighbors=9)
```

4.9 Check the Accuracy and Outputs



```
y_pred = model.predict(xtest)

from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(ytest, y_pred))
print(confusion_matrix(ytest, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.96      0.91      1605
           1       0.71      0.38      0.50       395

    accuracy                           0.85      2000
   macro avg       0.79      0.67      0.70      2000
weighted avg       0.83      0.85      0.83      2000

[[1544   61]
 [ 245  150]]
```

```
model.score(xtest, ytest)*100
```

```
84.7
```

```
import numpy as np
data_new = [1, 22, 3, 12000.05, 2, 1, 1, 48000.02, 0, 1, 0]
data_new = np.array(data_new).reshape(1,-1)
model.predict(data_new)
```

```
array([0], dtype=int64)
```

Chapter 5

Reflection Notes

6.1 Skills acquired

1. Understand, evaluate, design and implement artificial intelligence models.
2. Implement contemporary artificial intelligence techniques, from knowledge representation, to deep learning, developing in demand skills and leadership qualities for an exciting career in AI.
3. Apply the legal, ethical, social and philosophical context for practical AI projects.
4. Extend knowledge in artificial intelligence through research, experimentation and analysis.
5. Practical or hands on experience in training an ML model.
6. Gain expertise in technical drawing to visualize concepts.

6.2 Technical Outcomes

- Machine learning involves computations on large data sets, hence we learnt strong basic fundamental knowledge such as computer architecture, algorithms and data structure complexity. Getting in depth into the python language and exploring new commands.
- Synthesize visual perception skills along with drawing skills to visually communicate ideas. Deconstruction of designs for its motives and inspirations. To learn to synthesize data and make connections within the data points using the available frameworks.
- To frame an appropriate actionable problem statement with reference to user needs and contextual alignments.
- Data analysis of different data sets and to understand the concepts on a real world basis to implement and make use of AI/ML in our upcoming career.
- To train different models and to make sure the requirement of the respective clients and make to implement a model according to their requirements.

6.3 Time Management

Time management helps you allocate time for the most important tasks. When we follow a schedule we don't have to spend time and energy on what to do. Instead we can focus on what matters and do well. The quality of the work will suffer if we are constantly worrying about meeting the deadlines. Time management helps to prioritize the tasks, so we can have enough time to focus on each project to put in the effort and produce high quality outcomes.

Many software companies have to work against tight timelines. Proper time management will allow us to allocate enough time to meet each deadline. Planning ahead also keeps us calm and think freely to work more in an efficient way.

6.4 Personality Development

Personality development is referred to as a process of developing and enhancing one's personality. It helps one to gain confidence and high self esteem. It is essential to think positive and don't get upset over minor things, to be a little flexible and always look at the broader perspectives of life. Do not think of harming others and share whatever you know. Always help others. Be a patient listener and never interrupt when others are speaking. Try to imbibe good qualities of others.

Confidence is the key to a positive personality. Exude confidence and positive aura wherever you go. Personality development teaches you to be calm and composed even at stressful situations. Never over react. Avoid finding faults in others. Learn to be a little broad minded and flexible.

Chapter 6

Conclusion

Companies can have a clear view and can provide them some exiting offers to stay in that service. The obtained results show that our proposed churn model produced better results and performed better by using machine learning techniques.

In upcoming time it is necessary to reduce further more features in order to obtain better accuracy and introducing some more machine learning models for better performance.

In conclusion, this internship has been a very useful experience for me. I can safely say that my understanding of the job environment has increased greatly. However, I do think that there are some aspects of the job that I could have done better and that I need to work on.

I have built more confidence in usage of software tools. The two main things I learnt after my experience in this firm are time management and being self-motivated. I have gained new knowledge and skills and met new people. Usage of big data tools can improve operational efficiency.

Data analysis helps companies make informed decisions, create a more efficient marketing strategies, improve customer experience, streamline operations , among many other things. Usage of charts, maps, other visual representations of data to help present your findings in an easy-to-understand way. Improving the data visualization skills often means learning visualization software.

Chapter 7

References

- C. Blank and T. Hermansson, "A Machine Learning approach to churn prediction in a subscription based service," KTH, Stockholm, 2018.
- D. Buö and M. Kjellander, "Predicting Customer Churn at a Swedish CRM-system Company," Linköpings Universitet, Linköping, 2014.
- K. Mishra and R. Rani, "An inclusive survey on machine learning for CRM: a paradigm shift," in 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017.
- H. Gebert, M. Geib, L. Kolbe and W. Brenner, "Knowledge-enabled customer relationship management: integrating customer relationship management and knowledge management concepts," Journal of Knowledge Management, vol. 7, no. 5, pp. 107-123, 2003.
- M. Sergue, "Customer Churn Analysis and Prediction using Machine Learning for a B2B SaaS company," KTH, Stockholm, 2020
- D. L. Garcia, A. Nebot and A. Vellido, "Intelligent data analysis approaches to churn as a business problem: a survey," Knowledge and Information Systems, vol. 51, no. 3, pp. 1-56, 2017.
- N. Gordini and V. Veglio, "Customers churn prediction and marketing retention strategies. An application of support vector machines based on the AUC parameter-selection technique in B2B e-commerce industry," Industrial Marketing Management, vol. 62, pp. 100-107, 2017.