PROJECT - 5

PYTHON: TELECOM CASE STUDY - SEGMENTATION-CLASSIFICATION

ABSTRACT:

The goal of this study is to apply analytical techniques to predict a customer churn and analyse the churning and non-churning customers by using data from an internet connection company.
Like most companies that supply goods and services over the internet, this company mainly deals with customer remotely.
This makes it difficult to determine whether customer is satisfied with company or not.

Company tries to understand about the churn probability of the customer based on which incentive will be provided to the customer.

SUMMARIES OF PROBLEM, DATA, METHODS, AND TECHNOLOGIES:

❖ PROBLEM SUMMARY

Customer is looking to get a clear and early indication of customers which are likely to churn, when and why, so that right offer can be served at right time.

The project is divided into 3 pieces namely –

A> Data Preparation of active customers
B> Data Preparation of churned customers
C> Modelling the prepared data to get desired outcome

❖ DATA SUMMARY

The input data provided is in CSV data format. The data need to be imported using 'read_csv' function of 'pandas' library.

The size of the data was moderate; however the variability of data was extensive.

There are large numbers of categorical column with high cardinality.

Provided below is the **data dictionary** of Active and churned customers:-

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90
Linkedin: https://www.linkedin.com/in/kaustav-banerjee-584525149

DATA DICTIONARY FOR CHURNED CUSTOMERS:

Fields/ Variables	Descriptions	Type	Var_length
Site_account_number	Account # (Unique key)	char	14
ACCOUNT_STATUS	Present Account Status	char	6
First_communication_date		num	8
CHURN_DATE		num	8
Upgrade_date	Date of Upgradation	char	12
SERVICE_PLAN	Plan Used	char	13
MODEL		char	7
SALES_CHANNEL	Channel through which the customer was acquired	char	15
DIRECT_INDIRECT_CHANNEL	Channel Sub Classification	char	9
SALES_SOURCE		char	51
COMPANY_SOURCE_NAME		char	13
FIRST_NAME	Account holder name	char	14
LAST_NAME		char	15
ADDRESS1	Account holder address	char	62
CITY		char	19
STATE		char	2
ZIPCODE		num	8
EMAIL_ADDRESS	Contact info	char	48
EMAIL_STATUS	Responds to emails/ feedbacks	char	19
SATELLITE		char	13
SITE_TYPE_DESC		char	16
WARRANTY_NAME		char	59
MOST_RECENT_SALES_CHANNEL		char	26
Gender	1 : Male 2 : Female	num	8
Date_of_Birth	DOB of Account holder	char	13
Age	Age of Account holder	num	8
Income	Income of Account holder (Refer Income table)	char	7
Marital_Status	0 - Single 1 - Married 2 - Divorced	char	7
Presence_of_children	0 - No children 1 - Atleast 1 child present	char	7
Computer_owner	Y - Has computer N - No computer	char	7

DATA DICTIONARY FOR ACTIVE CUSTOMERS:

Fields/ Variables	Descriptions	Type	Var_length
Site_account_number	Account # (Unique key)	char	14
ACCOUNT_STATUS	Present Account Status	char	9
First_Communication_date	First date of communication	char	12
Upgrade_date	Date of Upgradation	char	12
SERVICE_PLAN	Plan Used	char	13
MODEL		char	7
SALES_CHANNEL	Channel through which the customer was acquired	char	15
DIRECT_INDIRECT_CHANNEL	Channel Sub Classification	char	9
SALES_SOURCE		char	51
COMPANY_SOURCE_NAME		char	13
FIRST_NAME	Account holder name	char	14
LAST_NAME		char	15
ADDRESS1	Account holder address	char	62
CITY		char	19
STATE		char	2
ZIPCODE		num	8
EMAIL_ADDRESS	Contact info	char	48
EMAIL_STATUS	Responds to emails/ feedbacks	char	19
SATELLITE		char	13
SITE_TYPE_DESC		char	16
WARRANTY_NAME		char	59
MOST_RECENT_SALES_CHANNEL		char	26
Gender	1 : Male 2 : Female	num	8
Date_of_Birth	DOB of Account holder	char	13
Age	Age of Account holder	num	8
Income	Income of Account holder (Refer Income table)	char	7
Marital_Status	0 - Single 1 - Married 2 - Divorced	char	7
Presence_of_children	0 - No children 1 - Atleast 1 child present	char	7
Computer_owner	Y - Has computer N - No computer	char	7

Both the data frames have same number of columns except “CHURN_DATE” which is unique to churned customers’ data frame only.

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

❖ METHODS SUMMARY

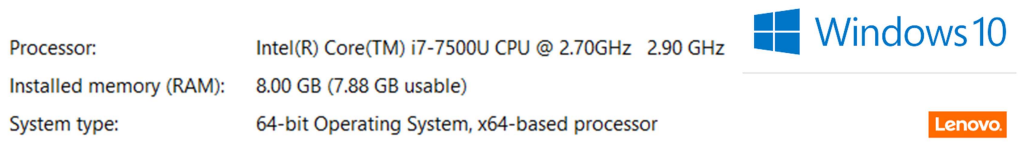
Table shows the wide variety of data pre-processing, analysis, and visualization techniques that I applied to complete the tasks as part of the project –

Task Details	Analytical Techniques	Visualization Techniques
Data Manipulation , cleaning and preparation for active and churned customers'	Descriptive statistics Straightforward data manipulation	pandas_profiling.ProfileReport barplot heatmap
Data Modelling and segmentation	Decision Tree and Random Forest	

❖ TECHNOLOGIES SUMMARY

The following list summarizes the technology that I used:

Computing platforms:

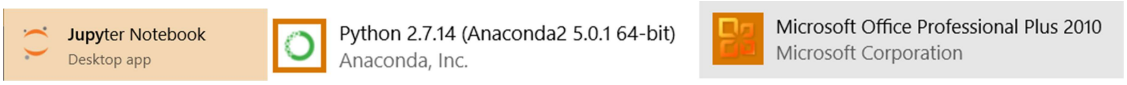


❖ DATA PRE-PROCESSING AND DATA MODELLING:

Microsoft Excel 2010 on the single-machine platform

Jupyter Notebook on the single-machine platform

Python 2.7.14 (Anaconda2 5.0.1 64bit on the single-machine platform



Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

SUMMARY OF DATA PROCESSING AND MODELING TECHNIQUE:

❖ DATA PROCESSING:

One of the major challenging portions of this project was data cleaning and data preparation which includes:-

A. Dropping variables with high cardinality and non-retrospective

```
telecom_active.drop(['ACCOUNT_STATUS' , 'ADDRESS', 'CITY', 'COMPANY_SOURCE_NAME',
'DATE_OF_BIRTH', 'EMAIL_ADDRESS', 'FIRST_COMMUNICATION_DATE', 'FIRST_NAME',
'LAST_NAME', 'ZIPCODE','UPGRADE_DATE','STATE','SALES_SOURCE'], axis=1, inplace = True)
```

B. Replacing NULL values appropriately in categorical and numerical variables and Missing value treatment (Sample Snippet)

```
telecom_active[['COMPUTER_OWNER', 'DIRECT_INDIRECT_CHANNEL', 'FEEDBACK',
'SALES_CHANNEL' ]] = telecom_active[['COMPUTER_OWNER', 'DIRECT_INDIRECT_CHANNEL',
'FEEDBACK', 'SALES_CHANNEL' ]].replace(np.nan, 'NA', regex=True)

telecom_active[['SERVICE_ PLAN', 'MODEL', 'SATELLITE', 'WARRANTY_NAME',
'MOST_RECENT_SALES_CHANNEL' ]] = telecom_active[['SERVICE_ PLAN', 'MODEL',
'SATELLITE', 'WARRANTY_NAME', 'MOST_RECENT_SALES_CHANNEL']].replace('#REF!', 'NA',
regex=True)
```

C. Squeezing the columns of similar groups together to reduce cardinality and trimming the unwanted spaces

COLUMN NAME - MOST_RECENT_SALES_CHANNEL

```
- Creating new categorical variable MOST_RECENT_SALES_CHANNEL_CAT
- Replacing "DW6000 Upgrade" with "DW6K Upgrade", "DW7000 Upgrade" with "DW7K Upgrade"
- Replacing "HN7000 Upgrade", "HN9000 Upgrade" with "HN9K Upgrade"
```

(Sample code Snippet)

```
telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']] =
telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']].replace(["DW6000 Upgrade"],
"DW6K Upgrade", regex=True)

telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']] =
telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']].replace(['DW7000 Upgrade'], 'DW7K
Upgrade', regex=True)

telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']] =
telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']].replace(['HN7000 Upgrade'], 'HN7K
Upgrade', regex=True)

telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']] =
telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']].replace(['HN9000 Upgrade'], 'HN9K
```

```
Upgrade', regex=True)

telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']] =
telecom_active[['MOST_RECENT_SALES_CHANNEL_CAT']].replace(['0','UNKNOWN'], 'NA',
regex=True)
```

D. Creating indicators for churn and upgrade indicators

- CHURN_IND 0 implying NOT churned, 1 implying churned

- UPGRADE_IND 0 implying NOT upgraded, 1 implying upgraded

By performing rigorous data manipulation and cleaning both the churn and active dataset were brought to identical format for merging into single dataset –

```
SITE_ACCOUNT_NUMBER
SALES_CHANNEL
DIRECT_INDIRECT_CHANNEL
FEEDBACK
SITE_TYPE_DESC
GENDER
AGE
INCOME
MARITAL_STATUS
PRESENCE_OF_CHILDREN
COMPUTER_OWNER
SERVICE_PLAN_CAT
MODEL_CAT
MOST_RECENT_SALES_CHANNEL_CAT
SATELLITE_CAT
WARRANTY_NAME_CAT
CHURN_IND
UPGRADE_IND
```

❖ **DATA MODELING:**

A. Encoding data because of large number of categorical variables

Because of high variability and cardinality of data in multiple columns I used Label Encoder to transform categorical variable to encoded numerical variables

(Code Snippet)

```
# Encoding Categorical Features
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()

telecom_data['SALES_CHANNEL_EN'] = encoder.fit_transform(telecom_data['SALES_CHANNEL'])
telecom_data['DIRECT_INDIRECT_CHANNEL_EN'] = encoder.fit_transform(telecom_data['DIRECT_INDIRECT_CHANNEL'])
telecom_data['FEEDBACK_EN'] = encoder.fit_transform(telecom_data['FEEDBACK'])
telecom_data['SITE_TYPE_DESC_EN'] = encoder.fit_transform(telecom_data['SITE_TYPE_DESC'])
telecom_data['INCOME_EN'] = encoder.fit_transform(telecom_data['INCOME'])
telecom_data['COMPUTER_OWNER_EN'] = encoder.fit_transform(telecom_data['COMPUTER_OWNER'])
telecom_data['SERVICE_PLAN_CAT_EN'] = encoder.fit_transform(telecom_data['SERVICE_PLAN_CAT'])
telecom_data['MODEL_CAT_EN'] = encoder.fit_transform(telecom_data['MODEL_CAT'])
telecom_data['MOST_RECENT_SALES_CHANNEL_CAT_EN'] = encoder.fit_transform(telecom_data['MOST_RECENT_SALES_CHANNEL_CAT'])
telecom_data['SATELLITE_CAT_EN'] = encoder.fit_transform(telecom_data['SATELLITE_CAT'])
telecom_data['WARRANTY_NAME_CAT_EN'] = encoder.fit_transform(telecom_data['WARRANTY_NAME_CAT'])
```

B. Creating model using Decision Tree/random Forest

I used Decision tree to create model.

By using grid search best parameter was found to be –

{'max_depth': 15, 'max_features': 7}

I build the final model using RANDOM FOREST with 98% train accuracy and 96% test accuracy

TRAIN DATASET Prediction accuracy -

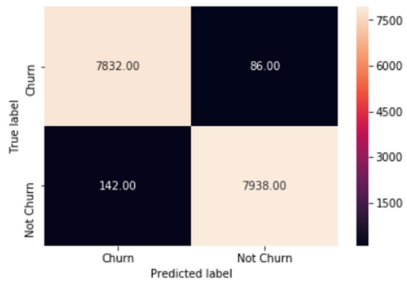
```
metrics.accuracy_score( radm_train_pred.actual, radm_train_pred.predicted )
```

```
0.9857482185273158 7
```

```
tree_cm_tr = metrics.confusion_matrix( radm_train_pred.predicted,
                                       radm_train_pred.actual,
                                       [1,0] )
sn.heatmap(tree_cm_tr, annot=True,
         fmt='.2f',
         xticklabels = ["Churn", "Not Churn"] , yticklabels = ["Churn", "Not Churn"] )

plt.ylabel('True label')
plt.xlabel('Predicted label')
```

```
Text(0.5,15,u'Predicted label')
```



TEST DADASET Prediction accuracy –

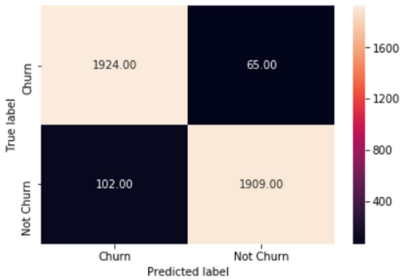
```
metrics.accuracy_score( radm_test_pred.actual, radm_test_pred.predicted )
```

```
0.9582500000000005
```

```
tree_cm = metrics.confusion_matrix( radm_test_pred.predicted,
                                    radm_test_pred.actual,
                                    [1,0] )
sn.heatmap(tree_cm, annot=True,
         fmt='.2f',
         xticklabels = ["Churn", "Not Churn"] , yticklabels = ["Churn", "Not Churn"] )

plt.ylabel('True label')
plt.xlabel('Predicted label')
```

```
Text(0.5,15,u'Predicted label')
```



Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

C. Probability of Churn

I added the probability of prediction to each active customer dataset to get a hold of CHURN PROBABILITY OF CHURN Prediction:

```
predictions = radm_clf.predict(df_test[features])
probs = radm_clf.predict_proba(df_test[features])
display(predictions)
```

Using the probability of churn (prob_true) I derived CHURN BAND to segment active customers –

CHURN SEGMENTATION -

```
0 - 0.3   - LOW CHURN PROBABILITY
0.3 - 0.7 - MEDIUM CHURN PROBABILITY
0.7 - 1   - HIGH CHURN PROBABILITY
```

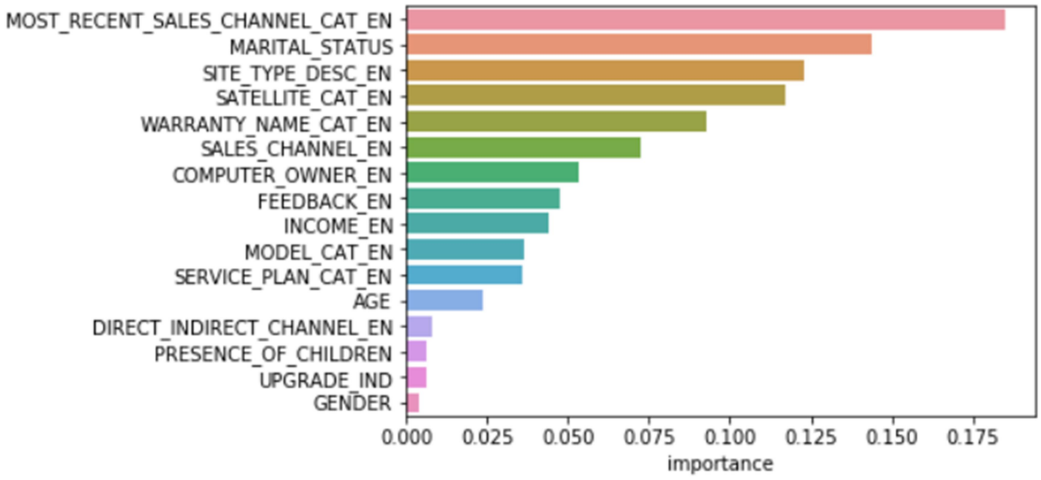
CHURN BAND	Count of SITE_ACCOUNT_NUMBER
HIGH_CHURN_PROBABILITY	321
LOW_CHURN_PROBABILITY	7351
MED_CHURN_PROBABILITY	2326
Grand Total	**9998**

```
def churntyp(df_telecom):
    if ((df_telecom.prob_true  >= 0.0) & (df_telecom.prob_true <= 0.3)):
        return 'LOW_CHURN_PROBABILITY'
    elif ((df_telecom.prob_true  > 0.3) & (df_telecom.prob_true <= 0.7)):
        return 'MED_CHURN_PROBABILITY'
    else:
        return 'HIGH_CHURN_PROBABILITY'
```

```
df_telecom['CHURN_BAND'] = df_telecom.apply(churntyp, axis=1)
```

UNDERSTANDING THE FEATURES THAT ARE INFLUENCING THE CHURN:

Using Random forest 'feature_importances' following features were found to be of highest importance



Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

Based on "feature_importances" ; we can recommend the incentive to customers

Results are available at – output/

Cleaned data frame for active customer - TELECOM_ACTIVE_CUSTOMER_CLEANED.xlsx
Cleaned data frame for active customer - TELECOM_CHURNED_CUSTOMER_CLEANED.xlsx
Active Data frame with churn probability - TELECOM_CHURNED_CUSTOMER_CLEANED.xlsx

Solution are available at – solution/

Data Manipulation for active customer - Telecom_DataCleaning_Active_Customer.ipynb
Data Manipulation for churned customer - Telecom_Model_Creation_prediction.ipynb
Model creation - Telecom_Model_Creation_prediction.ipynb

RECOMMENDATIONS

For customers with **HIGH_CHURN_PROBABILITY**

- Provide them a free upgrade or promotional offers/bonus to move to upgraded package.
- Ask them if they need a warranty, provide them offers or discount on buying warranty.

For customers with **MEDIUM_CHURN_PROBABILITY**

- Provide customers with loyalty/bonus/incentive points to boost confidence on company.
- Reach out to them asking for warranty purchase or upgrade.
- Ask for feedback on the Site and Satellite type.

General Recommendations –

a. Maximum churn is predicted where MOST_RECENT_SALES_CHANNEL is **Dealer-Generated Sale** or **HN7000 Upgrade**; these sales channel needs to be kept under scanner; keeping in mind they are also generating highest customers
b. **Spaceway** Site type is causing high churn probability- may be customers using the site type needs to be taken feedback on a regular interval.
c. Customers having "**SPACEWAY3**" and "**G17-HOR SERIES"** (G17-HOR-2K and G17-HOR-6K) satellite types needs to be changed or needs regular feedback; since it generation high churn rate.
d. Customers not having a **WARRANTY** have high churn rate. Give them a warranty plan; may be an offer for the purpose.
e. Look out for **BAD FEEDBACK** – it is one of the top reasons of churn.
f. Customers having annual income **less than $10K** or within **$60K – $80K** have high churn probability. Check for income statistics before providing service.
g. **HN7000S** model has high customer and high churn; be sure to take periodic reviews for the customer availing the model.

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90
Linkedin: https://www.linkedin.com/in/kaustav-banerjee-584525149

REFERENCES

https://github.com/

https://www.kaggle.com/

Analytixlabs Class 14-17 Case Study - HR Analytics and Telco Segmentation

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90
Linkedin: https://www.linkedin.com/in/kaustav-banerjee-584525149