American International University-Bangladesh (AIUB)

Final Project Report

Summer Semester [2021-2022]

Topic: *CUSTOMER CHURN PREDICTION*

Course: DATA WAREHOUSING AND DATA MINING
Section: *E*

Submitted By

Md.Miftahul Alam
18-38839-3

Chisty, Md Nomanuzzman

19-40431-1

Indrojit, Dhe Shaon

18-38674-3

Ahnaf, Sadman

16-32398-2

Submitted To

Dr. Akinul Islam Jony

Assistant Professor,
Computer Science

AIUB
Email: akinul@aiub.edu

Contents

Section 1: Project Overview

A lot of companies gain new and lose old customers every day, month, and year. But especially service-based companies who rely on subscription and monthly fees are concerned about having customers regularly taking their service. So, a term is used Customer Churn. Customer churn is the number of customers leaving your service within a year/certain time frame. Using Data mining techniques like Naïve Bayes, KNN Algorithm and Decision tree we are going to find out amounts of customers leaving the certain telecom company service and also see State-wise analysis of customer churn in USA. The whole project is done with the help of the WEKA tool.

Section 2: Dataset Overview

We collected our dataset from Kaggle. In 2020, a competition was held named “Customer Churn Prediction.” The participant analyzed the data and did deep learning work later using faulty test data to prove if their model is useful. Their training dataset was publicly available and useful for our project.

Url: *https://www.kaggle.com/competitions/customer-churn-prediction-2020/data*

It has a total of 4250 instances and 20 features/attributes. The total size of the dataset is 391.87 Kilobytes.

Feature Details:

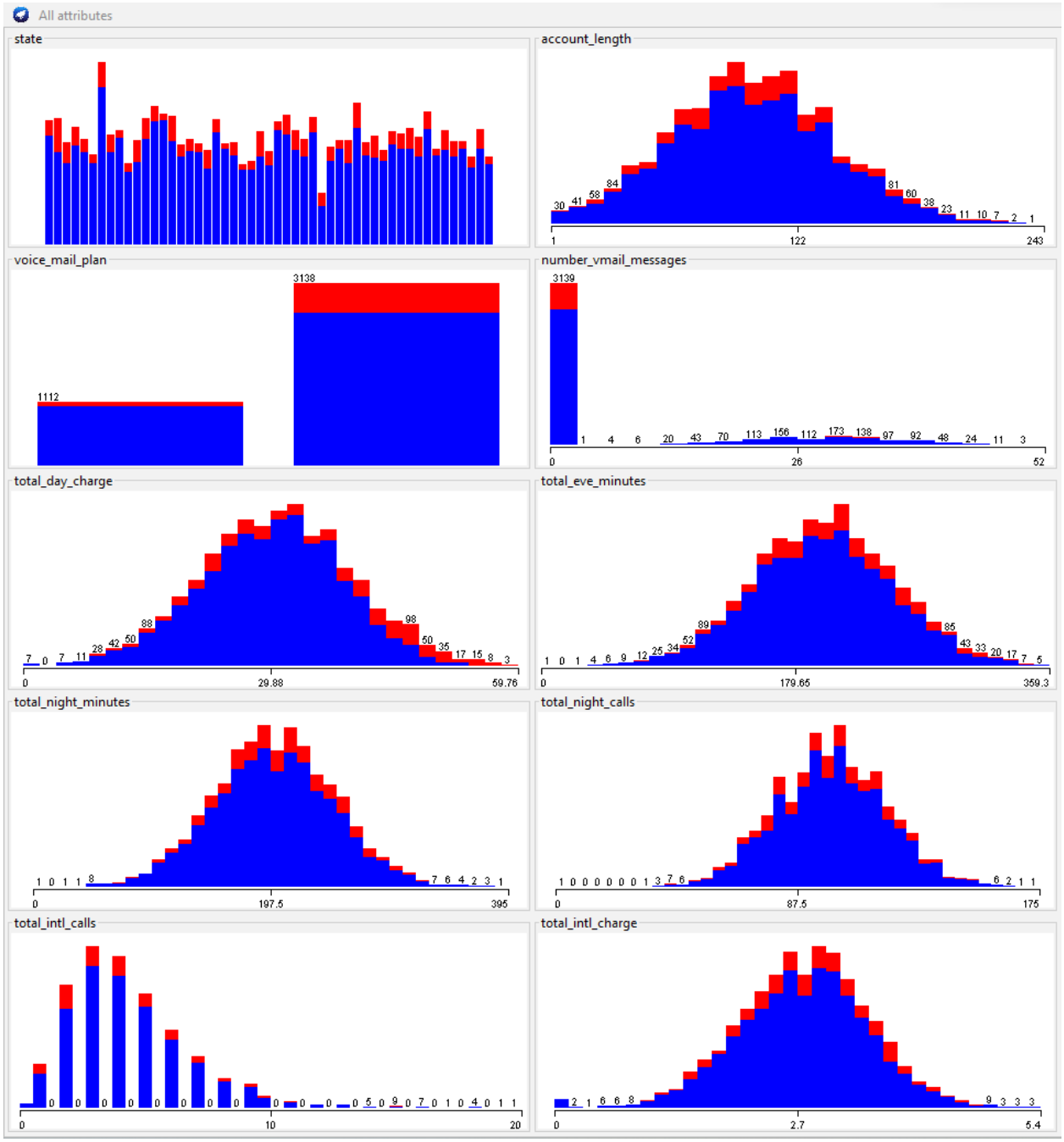
1. "state", *string*. 2-letter code of the US state of customer residence
2. "account_length", *numerical*. Number of months the customer has been with the current telco provider
3. "area_code", *string*="area_code_AAA" where AAA = 3 digit area code.
4. "international_plan", *(yes/no)*. The customer has international plan.

5. "voice_mail_plan", *(yes/no)*. The customer has voice mail plan.
6. "number_vmail_messages", *numerical*. Number of voice-mail messages.
7. "total_day_minutes", *numerical*. Total minutes of day calls.
8. "total_day_calls", *numerical*. Total minutes of day calls.
9. "total_day_charge", *numerical*. Total charge of day calls.
10. "total_eve_minutes", *numerical*. Total minutes of evening calls.
11. "total_eve_calls", *numerical*. Total number of evening calls.
12. "total_eve_charge", *numerical*. Total charge of evening calls.
13. "total_night_minutes", *numerical*. Total minutes of night calls.
14. "total_night_calls", *numerical*. Total number of night calls.
15. "total_night_charge", *numerical*. Total charge of night calls.
16. "total_intl_minutes", *numerical*. Total minutes of international calls.
17. "total_intl_calls", *numerical*. Total number of international calls.
18. "total_intl_charge", *numerical*. Total charge of international calls
19. "number_customer_service_calls", *numerical*. Number of calls to customer service
20. "churn", *(yes/no)*. Customer churn - target variable.

Here customer churn is our target variable and it’s our target class. It’s a binary classification since there are two values - Yes or No.

Section 3: Model Development

Firstly, we load the dataset properly using Weka preprocessing tool using the open file option and observed all the attributes and instances. Since the dataset was of good quality no missing values or discrepancies were not found.



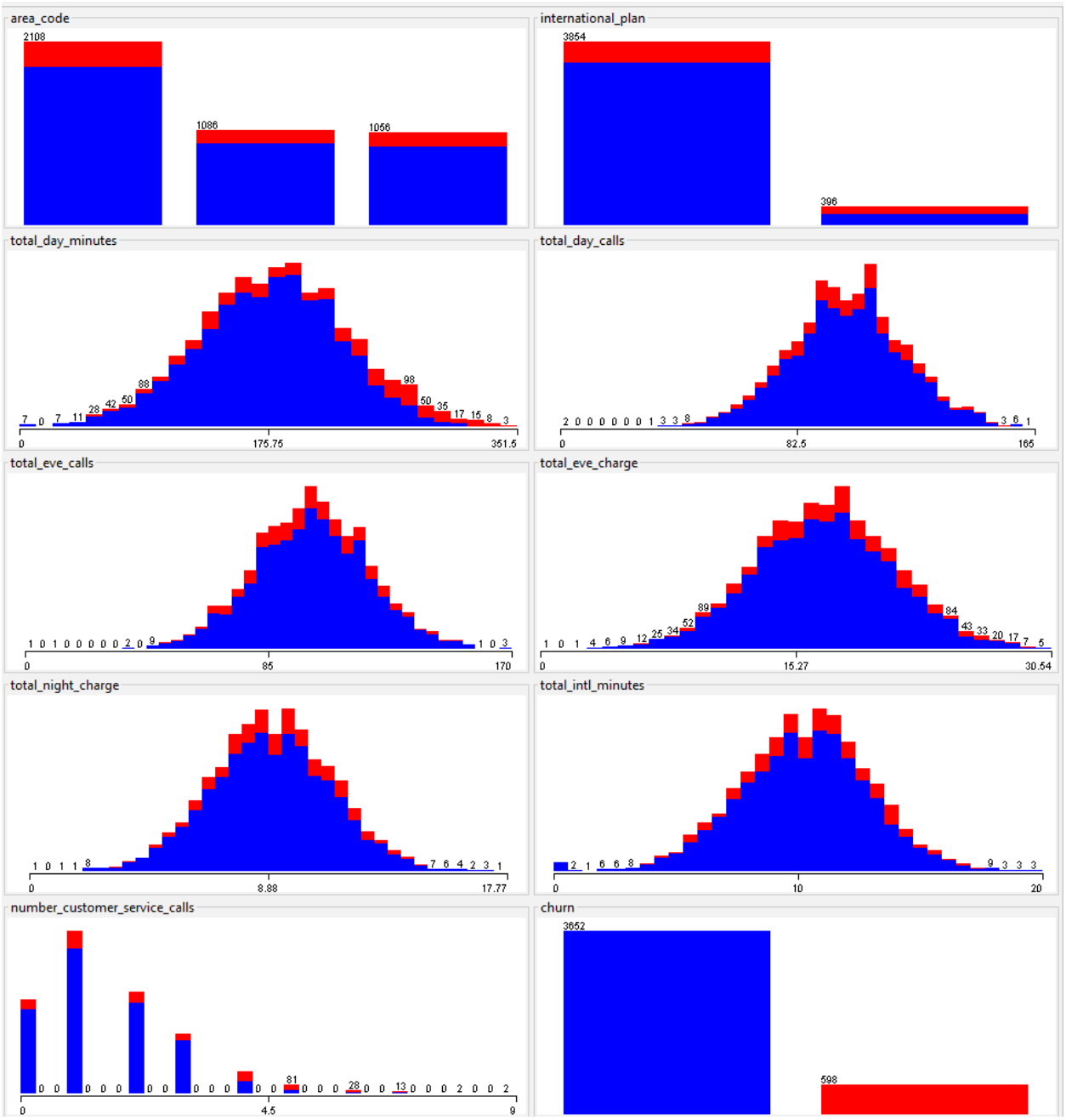


Figure: Depicting the Visualization of Attributes

Secondly, now comes the process of classification. We performed classification using three data mining techniques

1) Naïve Bayes

We selected the classify option. Then from the classifier option, we selected the Naïve Bayes from the Bayes category. We selected the 10-fold cross-validation technique since it provides the optimum result.

Area-wise customer churn analysis:

state		
OH	84.0	13.0
NJ	71.0	27.0
OK	63.0	17.0
MA	76.0	15.0
MO	71.0	11.0
LA	63.0	8.0
WV	121.0	20.0
IN	71.0	14.0
RI	82.0	7.0
IA	56.0	8.0
MT	64.0	18.0
NY	81.0	17.0
ID	95.0	13.0
VA	96.0	6.0
TX	80.0	20.0
FL	68.0	10.0
CO	72.0	10.0
AZ	71.0	8.0
SC	59.0	15.0
WY	86.0	11.0
HI	74.0	5.0
NH	69.0	11.0
AK	58.0	5.0
GA	58.0	8.0
MD	68.0	20.0
AR	61.0	12.0
WI	88.0	8.0
OR	85.0	16.0
MI	73.0	16.0
DE	68.0	14.0
UT	86.0	13.0
CA	30.0	11.0
SD	65.0	12.0
NC	74.0	8.0
WA	63.0	19.0
MN	90.0	20.0
NM	69.0	11.0
NV	67.0	18.0
DC	65.0	9.0
VT	77.0	11.0
KY	74.0	13.0
ME	74.0	17.0
MS	68.0	16.0
AL	89.0	14.0
NE	69.0	6.0
KS	73.0	16.0
TN	68.0	13.0
IL	74.0	7.0
PA	60.0	9.0

Here left side of the table shows the US state code with two words and the middle one shows No classification. The most-right side indicates the Yes classification. A number of customers left their telecom services.

```
=== Summary ===

Correctly Classified Instances        3774               88.8    %
Incorrectly Classified Instances       476               11.2    %
Kappa statistic                          0.4659
Mean absolute error                      0.1665
Root mean squared error                  0.2915
Relative absolute error                 68.8179 %
Root relative squared error             83.8384 %
Total Number of Instances             4250

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.961     0.557     0.913       0.961   0.936      0.477    0.839     0.956     no
               0.443     0.039     0.650       0.443   0.527      0.477    0.839     0.576     yes
Weighted Avg.  0.888     0.484     0.876       0.888   0.879      0.477    0.839     0.902
```

So this is the summary of the whole process outcome.

Correctly Classified Instances 3774 which is 88.8%

Incorrectly Classified instance is 476 which is 11.2%

2) KNN

Now we selected the IBK algorithm from the Lazy folders of classifiers. KNN in weka is known as IBK (Instance-based learning). The IBk algorithm does not build a model, instead, it generates a prediction for a test instance just-in-time. KNN is called a lazy learner because all it does calculating distances between the vectors and during training it does not require parametric guidance to form models. After selecting the IBK algorithm we will set the training requirements. We are going to use 10-fold cross-validation for good accuracy.

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3543               83.3647 %
Incorrectly Classified Instances       707               16.6353 %
Kappa statistic                          0.2136
Mean absolute error                      0.1665
Root mean squared error                  0.4078
Relative absolute error                 68.827  %
Root relative squared error            117.2666 %
Total Number of Instances             4250

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.928     0.741     0.884       0.928   0.906      0.218    0.593     0.883     no
               0.259     0.072     0.370       0.259   0.305      0.218    0.593     0.200     yes
Weighted Avg.  0.834     0.647     0.812       0.834   0.821      0.218    0.593     0.787
```

3543 instances were correctly classified which is around 83.3847%. 707 instances were incorrectly classified which is around 16.6353.

3)Decision Tree

Now comes the Decision tree. The decision tree is a decision support tool that uses a tree-like structured model of all possible decisions and consequences to reach an outcome from rigorous computations. After loading data, we went to classify options and then selected classifiers. There we found a tree folder. From that we selected J48. In Weka, a Decision tree is known as J48. Here we also used the 10-fold cross-validation technique for good accuracy.

Analyzing the dataset, we found the following conditions for constructing a decision tree:

1)total_day_minutes <248.6 and total_day_minutes >=248.6

2) If agreed with total_day_minutes then number_customer service calls<3. If not then total_day_minutes>=248.6.

3)international plan=no and international plan=yes.

4)total_international_calls<=2 and total_international_calls>2

5)total_international_minutes<=13 and total_international_minutes>13

6)voice_mail_plan=yes and no

7)total_day_minutes <=220.8 and total_ day_minutes>220.8

8)total eve_charge and total_day_charge.

The Decision tree we have constructed has 29 leaves and the size of the tree is 57.

```
total_day_minutes <= 248.6
|   number_customer_service_calls <= 3
|   |   international_plan = no
|   |   |   total_day_minutes <= 220.8: no (2775.0/68.0)
|   |   |   total_day_minutes > 220.8
|   |   |   |   total_eve_charge <= 22.7: no (388.0/24.0)
|   |   |   |   total_eve_charge > 22.7
|   |   |   |   |   voice_mail_plan = yes: no (7.0)
|   |   |   |   |   voice_mail_plan = no: yes (35.0/4.0)
|   |   international_plan = yes
|   |   |   total_intl_calls <= 2: yes (56.0)
|   |   |   total_intl_calls > 2
|   |   |   |   total_intl_minutes <= 13: no (201.0/3.0)
|   |   |   |   total_intl_minutes > 13: yes (47.0)
|   number_customer_service_calls > 3
|   |   total_day_minutes <= 160.2
|   |   |   total_day_minutes <= 134.7: yes (65.0)
|   |   |   total_day_minutes > 134.7
|   |   |   |   total_eve_minutes <= 232.5: yes (45.0/3.0)
|   |   |   |   total_eve_minutes > 232.5
|   |   |   |   |   total_night_calls <= 75: yes (2.0)
|   |   |   |   |   total_night_calls > 75: no (10.0/1.0)
|   |   total_day_minutes > 160.2
|   |   |   total_eve_charge <= 12.71
|   |   |   |   total_day_minutes <= 197.2: yes (10.0)
|   |   |   |   total_day_minutes > 197.2: no (17.0/7.0)
|   |   |   total_eve_charge > 12.71: no (154.0/27.0)
total_day_minutes > 248.6
|   voice_mail_plan = yes
|   |   international_plan = no: no (88.0/1.0)
|   |   international_plan = yes
|   |   |   total_intl_calls <= 5
|   |   |   |   total_intl_calls <= 2: yes (2.0)
|   |   |   |   total_intl_calls > 2: no (8.0)
|   |   |   total_intl_calls > 5: yes (4.0)
|   voice_mail_plan = no
|   |   total_eve_minutes <= 201
|   |   |   total_day_minutes <= 285.3
|   |   |   |   international_plan = no: no (104.0/13.0)
|   |   |   |   international_plan = yes
|   |   |   |   |   total_intl_calls <= 2: yes (5.0)
|   |   |   |   |   total_intl_calls > 2
|   |   |   |   |   |   total_intl_calls <= 7
|   |   |   |   |   |   |   total_eve_calls <= 85: yes (3.0/1.0)
|   |   |   |   |   |   |   total_eve_calls > 85: no (8.0)
|   |   |   |   |   |   total_intl_calls > 7: yes (2.0)
|   |   |   total_day_minutes > 285.3
|   |   |   |   total_eve_minutes <= 138.1
|   |   total_eve_minutes > 201
|   |   |   total_night_charge <= 7.75
|   |   |   |   total_day_minutes <= 277.7: no (30.0/13.0)
|   |   |   |   total_day_minutes > 277.7: yes (21.0)
|   |   |   total_night_charge > 7.75: yes (120.0/3.0)
```

```
Correctly Classified Instances        3997               94.0471 %
Incorrectly Classified Instances       253                5.9529 %
Kappa statistic                          0.7298
Mean absolute error                      0.09
Root mean squared error                  0.228
Relative absolute error                 37.2156 %
Root relative squared error             65.5675 %
Total Number of Instances             4250

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.983     0.318     0.950      0.983     0.966      0.737    0.885     0.958     no
               0.682     0.017     0.866      0.682     0.763      0.737    0.885     0.773     yes
Weighted Avg.  0.940     0.275     0.938      0.940     0.937      0.737    0.885     0.932
```

The model has successfully classified 3997 instances which are around 94.0471%. 253 instances were incorrectly classified which is around 5.95%.

Section 4: Discussion and Conclusion

Accuracy

Classifier	**Accuracy**
Naïve Bayes	*88.8%*
KNN	*83.384%*
Decision Tree	*94.0471%*

Confusion Matrix

Naïve Bayes

```
=== Confusion Matrix ===

    a    b   <-- classified as
 3509  143 |     a = no
  333  265 |     b = yes
```

KNN

```
=== Confusion Matrix ===

    a    b   <-- classified as
 3388  264 |     a = no
  443  155 |     b = yes
```

Decision Tree

```
=== Confusion Matrix ===

    a    b   <-- classified as
 3589   63 |     a = no
  190  408 |     b = yes
```

Predictive Accuracy

P=C/N

Naïve Bayes

P(N) = 3774 / 4250

= 0.888

KNN

P(K) = 354 3/ 4250

= 0.83

Decision Tree

P(D) = 3997 / 4250

=0.94.

We have clearly observed and calculated the performance of all three models namely: Naïve Bayes, KNN, and Decision tree. In terms of accuracy and precision Decision tree performs best. Its accuracy is around 94%. For this dataset and this problem analysis Decision tree performs well and also shows us a detailed analysis of the reasons which caused customers to leave the service and churn classification as yes. The major reasons found are international call quality, customer service response, and daytime calling hours. Costs during daytime vs Night time vs evening time also played role in the decision-making of customers to whether leave the service or not causing churn classification YES or No.

References:

1. Karvana, K. G. M., Yazid, S., Syalim, A., & Mursanto, P. (2019, October). Customer churn analysis and prediction using data mining models in banking industry. In *2019 International Workshop on Big Data and Information Security (IWBIS)* (pp. 33-38). IEEE.
2. Dalvi, P. K., Khandge, S. K., Deomore, A., Bankar, A., & Kanade, V. A. (2016, March). Analysis of customer churn prediction in telecom industry using decision trees and logistic regression. In *2016 symposium on colossal data analysis and networking (CDAN)* (pp. 1-4). IEEE.