CUSTOMER CHURN

Statistical analysis and model on factors influencing customer churn on an example Kaggle bank dataset.

PURDUE UNIVERSITY| KRANNERT SCHOOL OF MANAGEMENT
2021 SUMMER MODULE 2 | MGMT-58600-B03 – PYTHON

SU TIEN LEE | CHAYADEEPSAI CHERUKUPALLI
SRI MANOGNA GURIJALA | ALEJANDRO BRILLEMBOURG CUENCA

THE PROJECT

Choosing the dataset and defining project steps.

Customer Churn

- What is Customer Churn?
- Why is it important?
- How does it affect the banking industry?

Banking Dataset

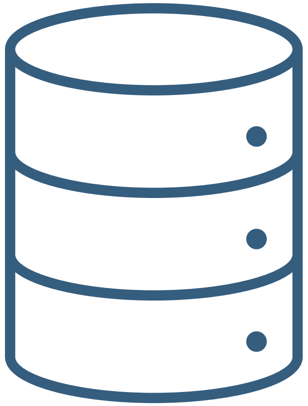
The Problem:

- A bank manager is concerned with customers leaving their credit card services.

The Case:

- Which features have greater impact on churn?
- Can we predict which customers will churn?
- How can the bank prevent churn?

Original Dataset



```
(10127, 23)
```

10127 rows

23 columns

```
False     10127
dtype: int64
```

0 duplicate rows

Original Dataset

```
['CLIENTNUM',
 'Attrition_Flag',
 'Customer_Age',
 'Gender',
 'Dependent_count',
 'Education_Level',
 'Marital_Status',
 'Income_Category',
 'Card_Category',
 'Months_on_book',
 'Total_Relationship_Count',
 'Months_Inactive_12_mon',
 'Contacts_Count_12_mon',
 'Credit_Limit',
 'Total_Revolving_Bal',
 'Avg_Open_To_Buy',
 'Total_Amt_Chng_Q4_Q1',
 'Total_Trans_Amt',
 'Total_Trans_Ct',
 'Total_Ct_Chng_Q4_Q1',
 'Avg_Utilization_Ratio',
 'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_
12_mon_1',
 'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_
12_mon_2']
```

23 column names

Original Dataset

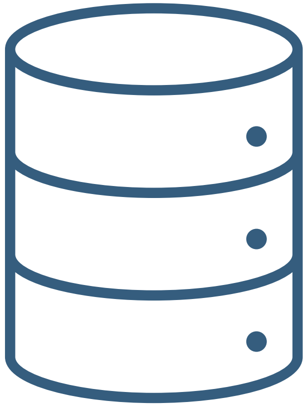
Feature	Description
Clientnum	Unique identifier for customers
Attrition_Flag	If the account is closed equals 1 else 0
Card_Category	Type of Card (Blue, Silver, Gold, Platinum)
Months_on_book	Months on book (Time of Relationship)
Total_Relationship_Count	Total no. of products held by the customer
Months_Inactive_12_mon	No. of months inactive in the last 12 months
Contacts_Count_12_mon	No. of Contacts in the last 12 months
Credit_Limit	Credit Limit on the Credit Card
Total_Revolving_Bal	Total Revolving Balance on the Credit Card
Avg_Open_To_Buy	Average of last 12 months
Total_Amt_Chng_Q4_Q1	Change in Transaction Amount (Q4 over Q1)
Total_Trans_Amt	Total Transaction Amount (Last 12 months)
Total_Trans_Ct	Total Transaction Count (Last 12 months)
Total_Ct_Chng_Q4_Q1	Change in Transaction Count (Q4 over Q1)
Avg_Utilization_Ratio	Average Card Utilization Ratio

Original Dataset

```
CLIENTNUM
0
Attrition_Flag
0
Customer_Age
0
Gender
0
Dependent_count
0
Education_Level
0
Marital_Status
0
Income_Category
0
Card_Category
0
Months_on_book
0
Total_Relationship_Count
0
Months_Inactive_12_mon
0
Contacts_Count_12_mon
0
Credit_Limit
0
Total_Revolving_Bal
0
Avg_Open_To_Buy
0
Total_Amt_Chng_Q4_Q1
0
Total_Trans_Amt
0
Total_Trans_Ct
0
Total_Ct_Chng_Q4_Q1
0
Avg_Utilization_Ratio
0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1    0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2    0
dtype: int64
```

0 null values
in each feature

Original Dataset



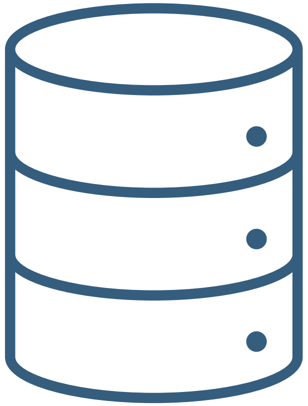
```
---------------  -----
 0   CLIENTNUM
10127 non-null   int64
 1   Attrition_Flag
10127 non-null   object
 2   Customer_Age
10127 non-null   int64
 3   Gender
10127 non-null   object
 4   Dependent_count
10127 non-null   int64
 5   Education_Level
10127 non-null   object
 6   Marital_Status
10127 non-null   object
 7   Income_Category
10127 non-null   object
 8   Card_Category
10127 non-null   object
 9   Months_on_book
10127 non-null   int64
 10  Total_Relationship_Count
10127 non-null   int64
 11  Months_Inactive_12_mon
10127 non-null   int64
 12  Contacts_Count_12_mon
10127 non-null   int64
 13  Credit_Limit
10127 non-null   float64
 14  Total_Revolving_Bal
10127 non-null   int64
 15  Avg_Open_To_Buy
10127 non-null   float64
 16  Total_Amt_Chng_Q4_Q1
10127 non-null   float64
 17  Total_Trans_Amt
10127 non-null   int64
 18  Total_Trans_Ct
10127 non-null   int64
 19  Total_Ct_Chng_Q4_Q1
10127 non-null   float64
 20  Avg_Utilization_Ratio
10127 non-null   float64
 21  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_
ive_12_mon_1  10127 non-null   float64
 22  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_
ive_12_mon_2  10127 non-null   float64
dtypes: float64(7), int64(10), object(6)
```

3 columns to drop

6 categorical features

14 numerical features

Original Dataset



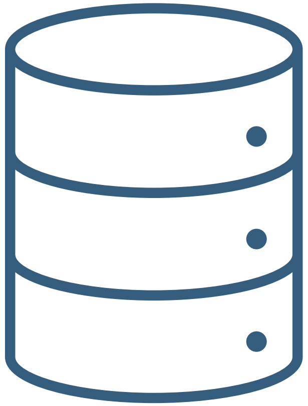
Top 5 Rows with Example Columns

Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category
Existing Customer	51	M	4	Unknown	Married	$120K +
Existing Customer	49	M	4	Uneducated	Single	80K−120K
Attrited Customer	48	M	2	Graduate	Married	60K−80K
Existing Customer	51	M	4	Uneducated	Single	80K−120K
Existing Customer	51	M	4	Graduate	Single	$120K +

.describe() with Example Columns

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon
count	1.012700e+04	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000
mean	7.391776e+08	46.325960	2.346203	35.928409	3.812580	2.341167	2.455317
std	3.690378e+07	8.016814	1.298908	7.986416	1.554408	1.010622	1.106225
min	7.080821e+08	26.000000	0.000000	13.000000	1.000000	0.000000	0.000000
25%	7.130368e+08	41.000000	1.000000	31.000000	3.000000	2.000000	2.000000
50%	7.179264e+08	46.000000	2.000000	36.000000	4.000000	2.000000	2.000000
75%	7.731435e+08	52.000000	3.000000	40.000000	5.000000	3.000000	3.000000
max	8.283431e+08	73.000000	5.000000	56.000000	6.000000	6.000000	6.000000

Original Dataset



3 Types of Categorical Features:

- Binary
- Ordinal
- Dummies

OUR PROCESS

EXPLORATORY DATA ANALYSIS

DATA PREPROCESSING

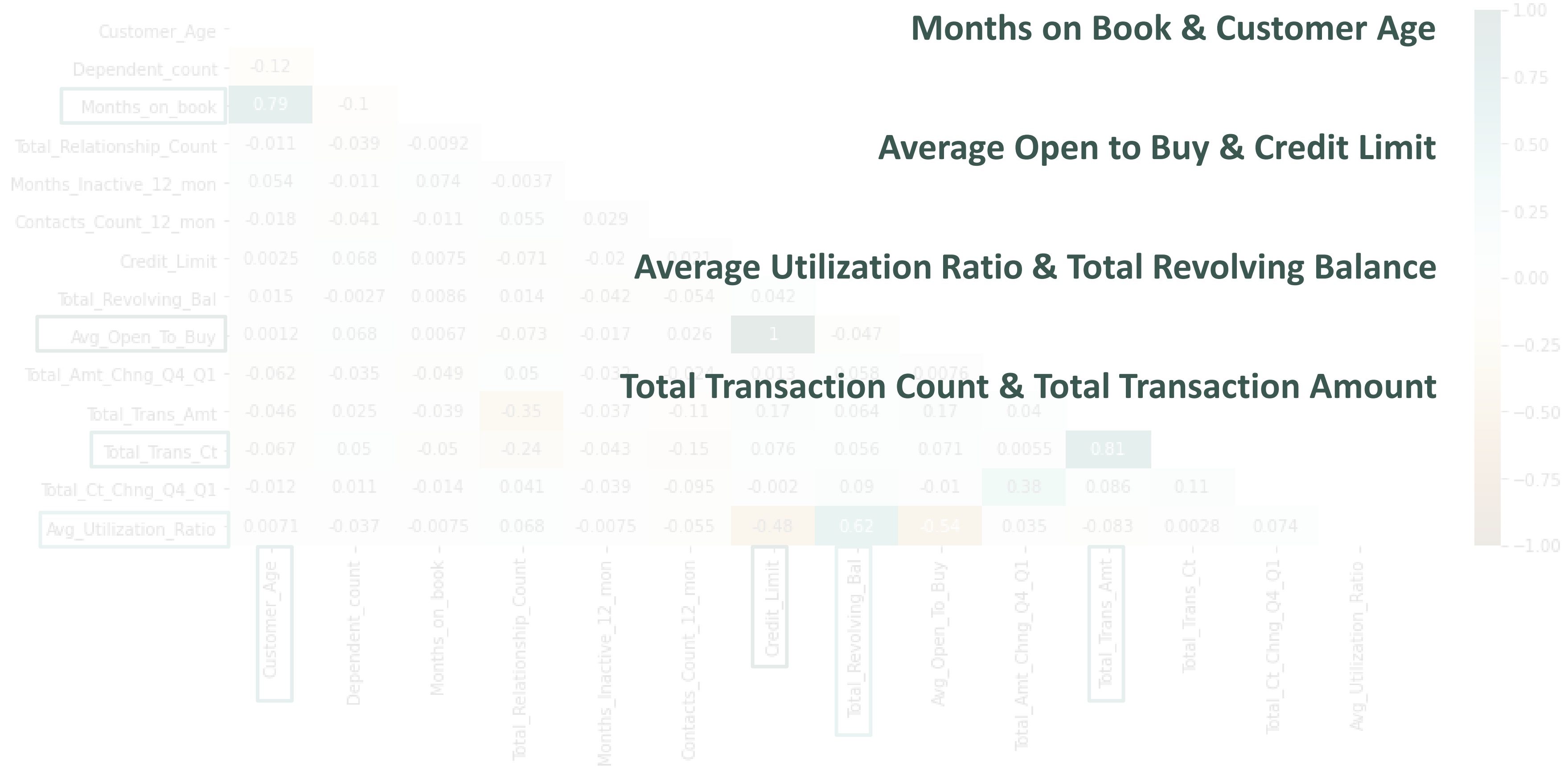
MODEL BUILDING

INSIGHTS

CASE STUDY DATASET

Exploratory analysis.

Triangle Correlation Heatmap



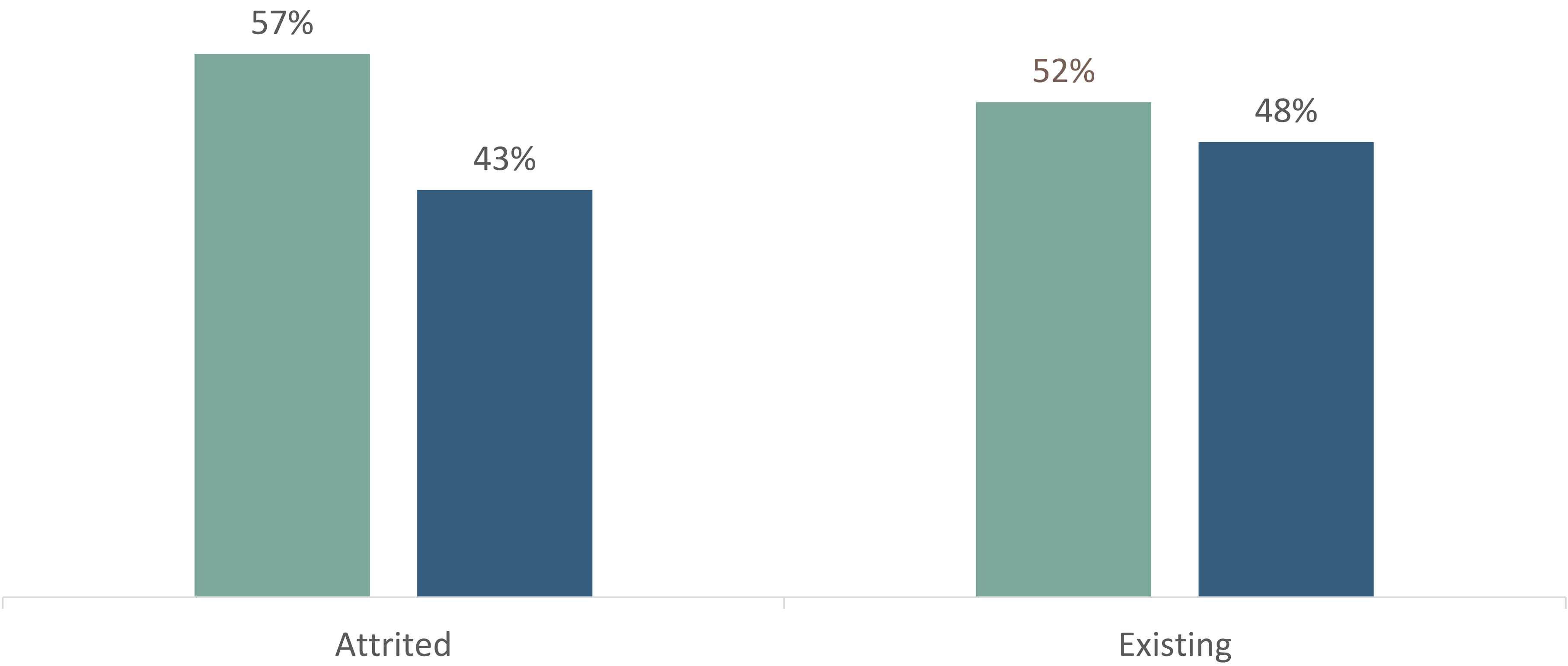
Months on Book & Customer Age

Average Open to Buy & Credit Limit

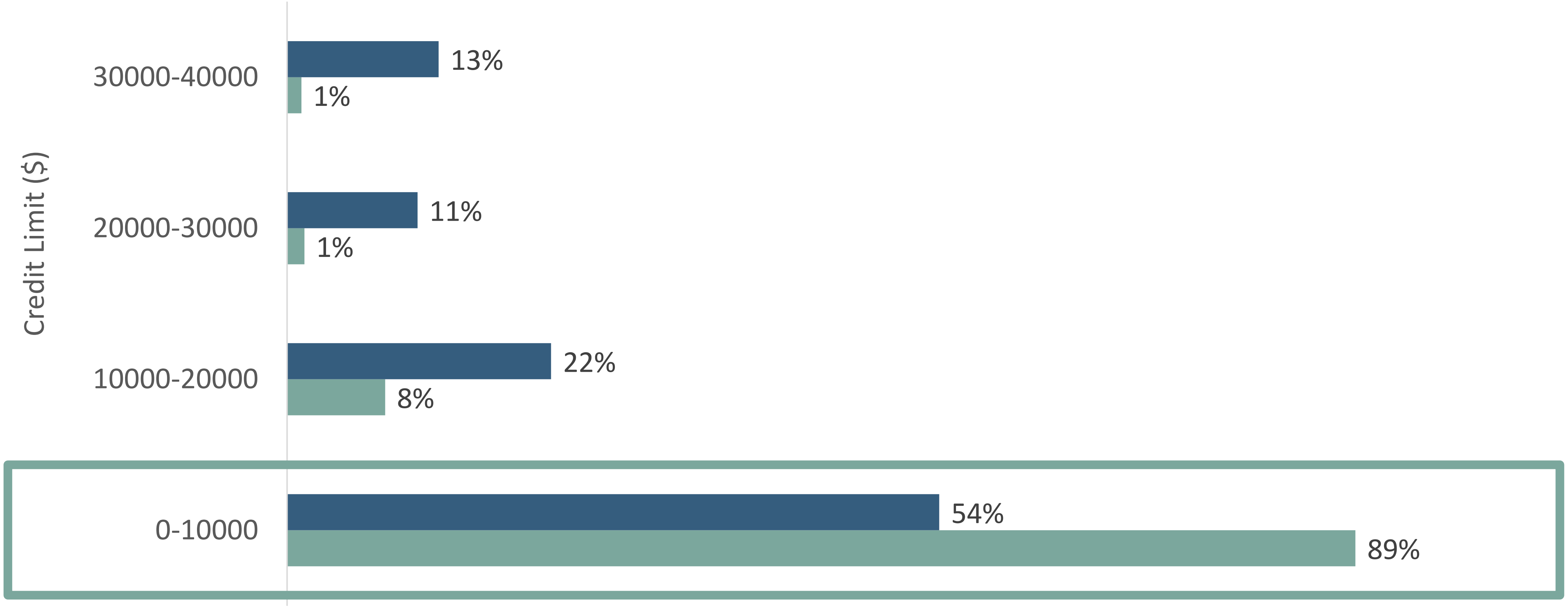
Average Utilization Ratio & Total Revolving Balance

Total Transaction Count & Total Transaction Amount

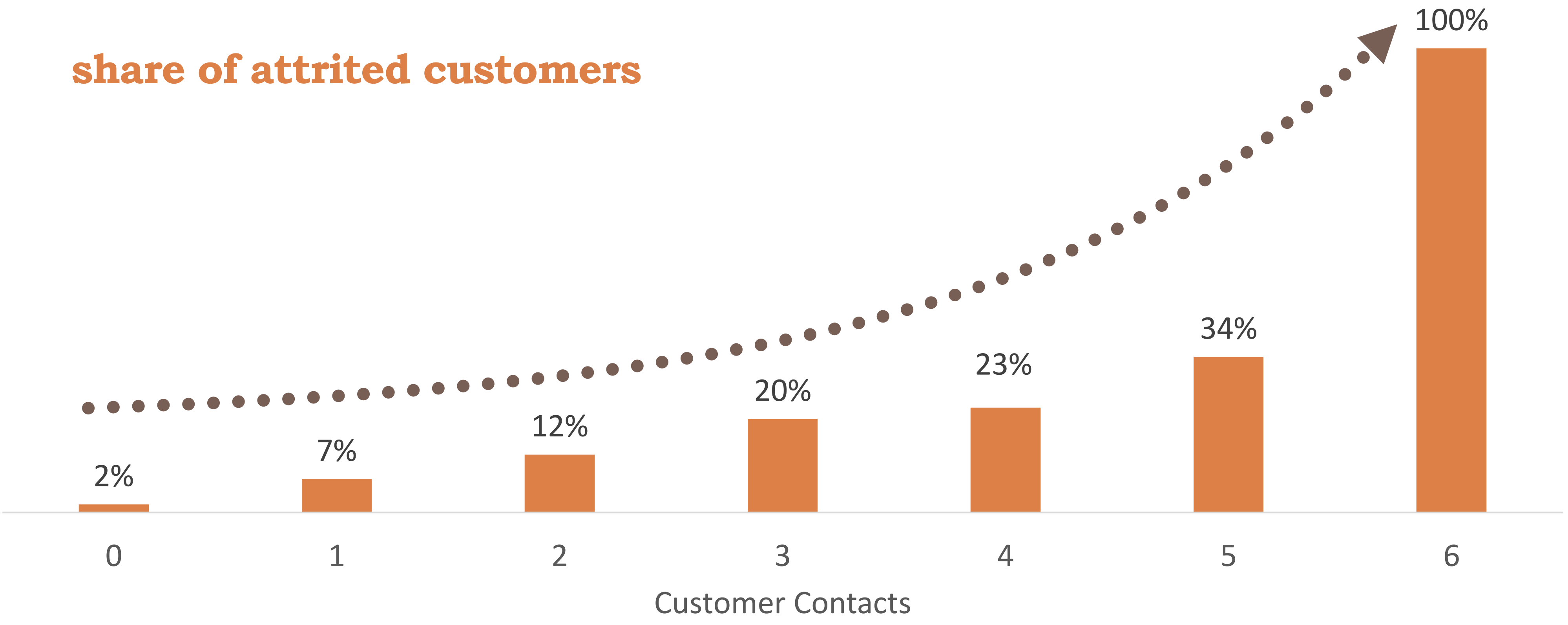
There are **4%** more existing **female customers** than **male customers**, yet **14%** more of the attrited customers are **female**.



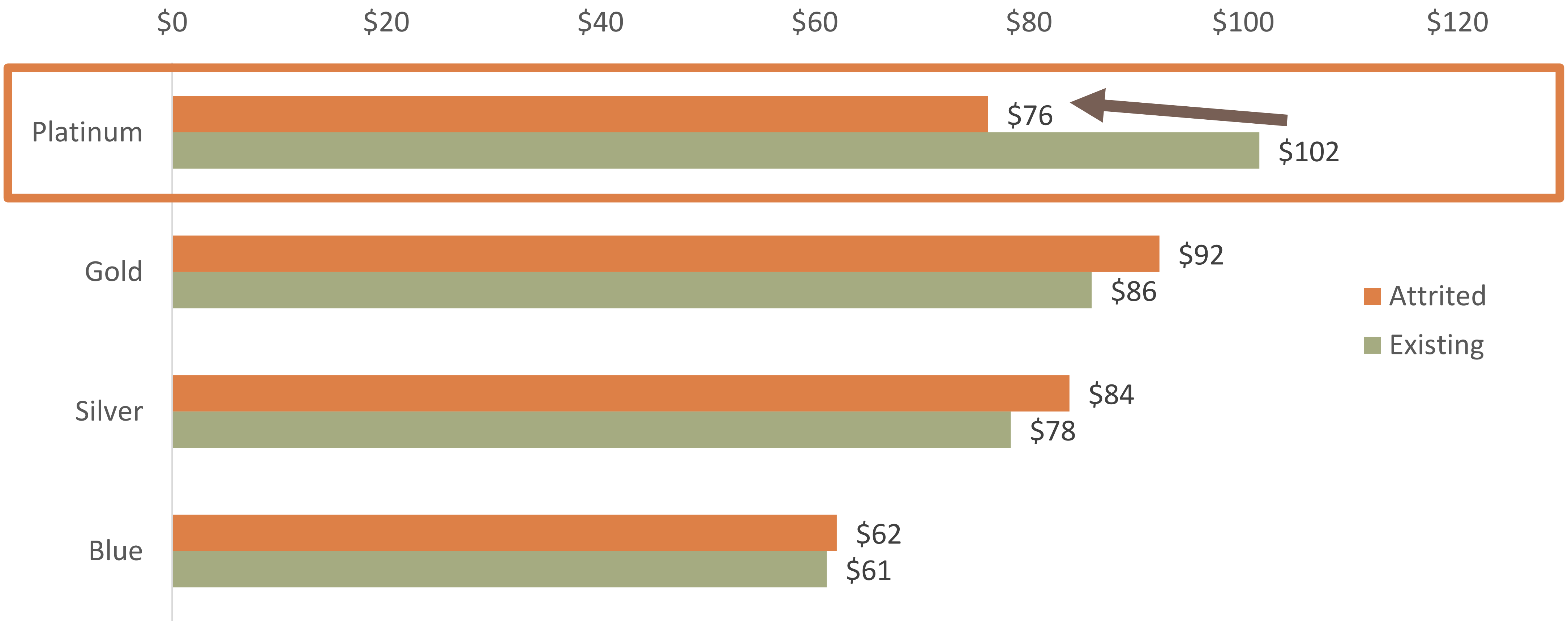
Unlike **male customers**, almost **90%** of **female customers** have **less than $10K** of credit limit.



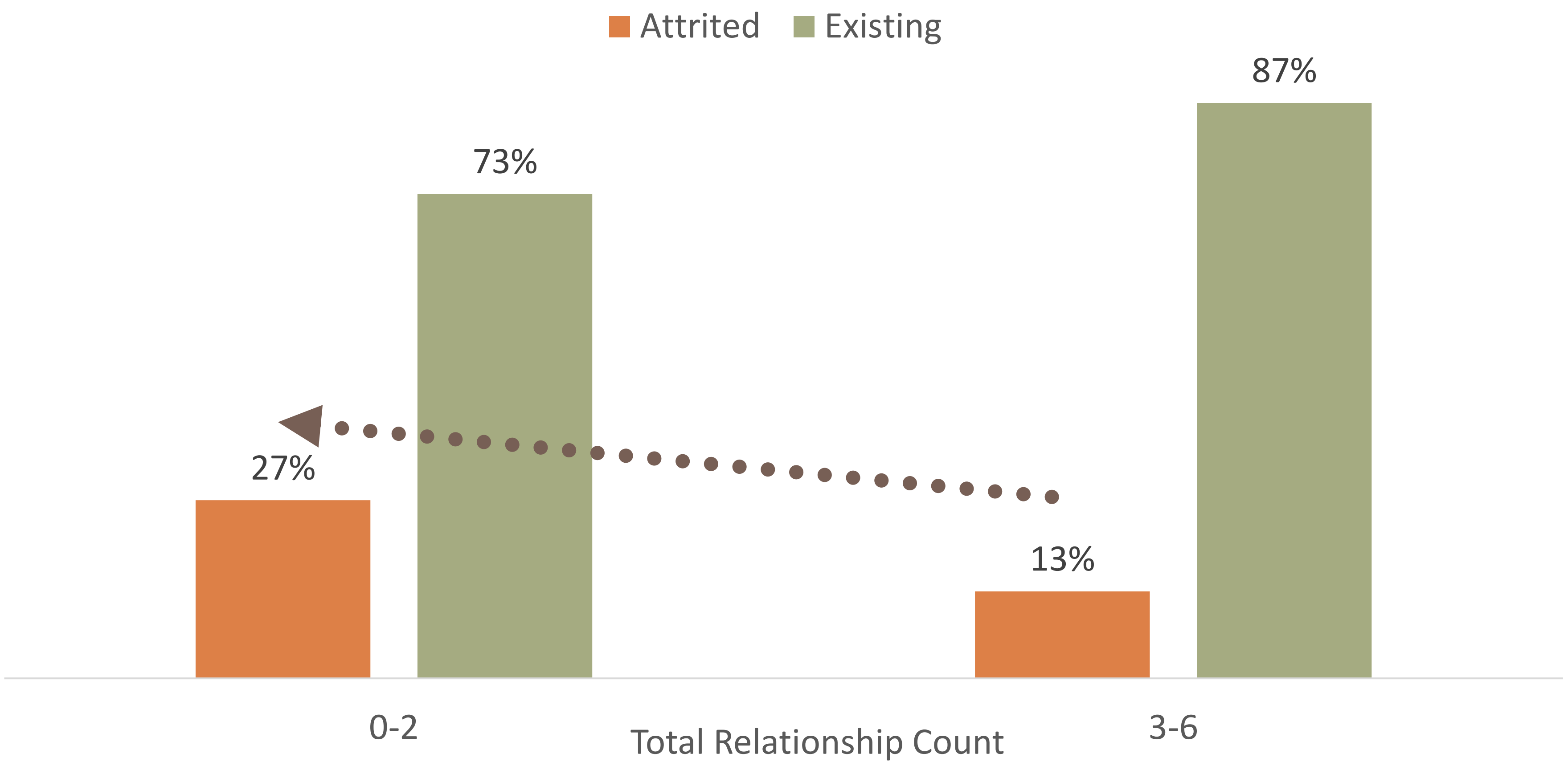
Every additional customer contact increases the likelihood to churn by **12%**.



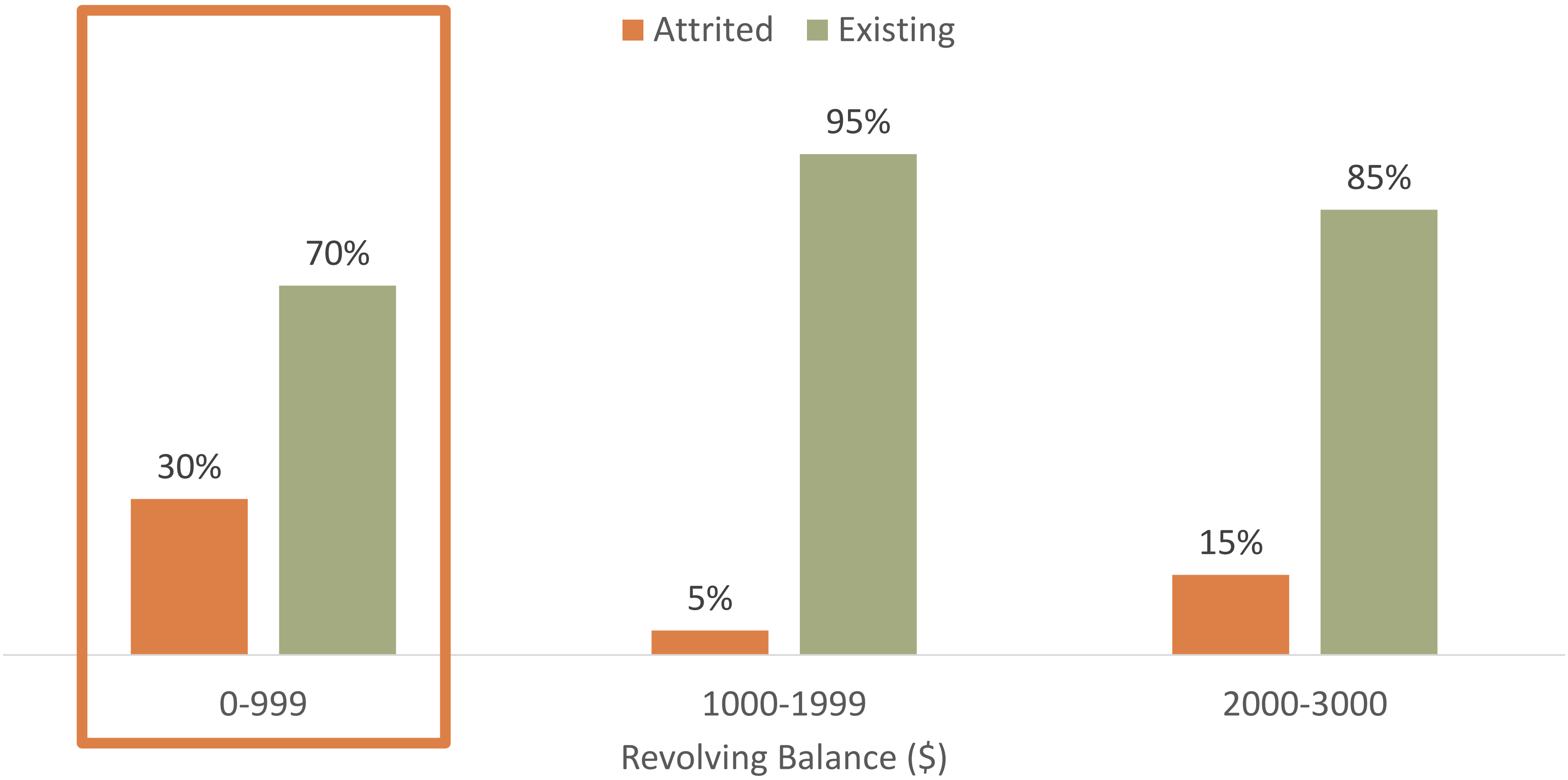
There is close to a **25%** drop in the average transaction amount for **attrited platinum card customers**.



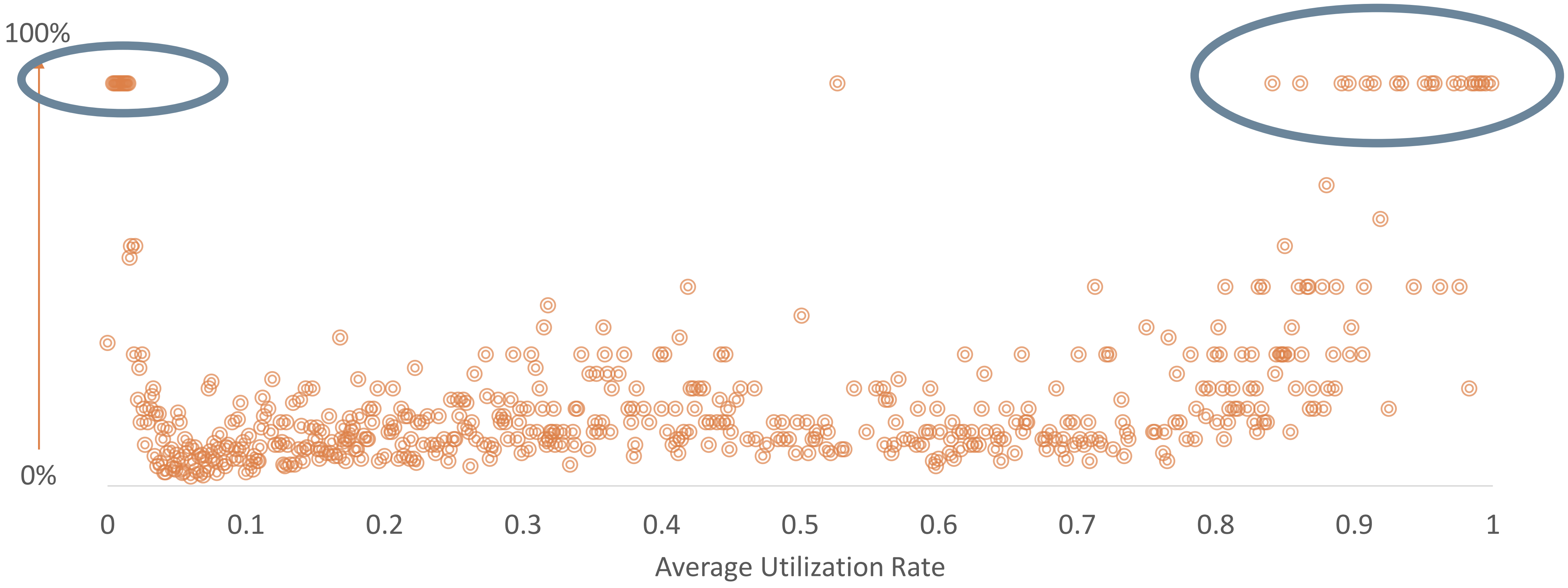
Customers with **less than 3** relationships are almost **2 times more likely** to churn.



30 out of 100 customers are likely to churn if their revolving balance is **below $1K**.



People at the extremes of **average utilization ratio** have higher percentages of **attrition rate**.



CASE STUDY DATASET

Data preprocessing and model building.

Original → Processed Data

3 Types of Categorical Features

- Binary Encoding
- Ordinal Encoding
- Encoding with Dummies

Original → Processed Data

Binary Encoding

```python
def enc_bin(df):
    '''the enc_bin function accepts a pandas dataframe and replaces
    categorical values with binary values'''
    cols = list(df.columns.values)

    if 'Gender' in cols:
        df['Gender'] = df['Gender'].replace({'F':1, 'M':0})
    else:
        None

    if 'Attrition_Flag' in cols:
        df['Attrition_Flag'] = df['Attrition_Flag'].replace({
            'Existing Customer':1, 'Attrited Customer':0})
    else:
        None

    return df
```

Original → Processed Data

Ordinal Encoding

```
def enc_ord(df):
    '''the enc_ord function accepts a pandas dataframe and replaces
    categorical values with ordinal values'''
    cols = list(df.columns.values)
    if 'Card_Category' in cols:
        ord_val = {'Card_Category': {'Blue':1, 'Silver':2,
                                     'Gold':3, 'Platinum':4}}
        df = df.replace(ord_val)
    else:
        None
    return df
```

Original → Processed Data

Dummies Encoding

```
def enc_dum(df):
    '''the enc_dum function accepts a pandas dataframe and creates
    dummy feature values for categorical features'''
    cols = list(df.columns.values)

    if 'Education_Level' in cols:
        df = pd.get_dummies(df, columns = ["Education_Level"],
                            prefix = ["EDU_LVL_"], drop_first = True)
    else:
        None

    if 'Marital_Status' in cols:
        df = pd.get_dummies(df, columns = ["Marital_Status"],
                            prefix = ["MAR_ST_"], drop_first = True)
    else:
        None

    if 'Income_Category' in cols:
        df = pd.get_dummies(df, columns = ["Income_Category"],
                            prefix = ["INC_CAT_"], drop_first = True)
    else:
        None

    return df
```

Original → Processed Data

Dummies Encoding Example

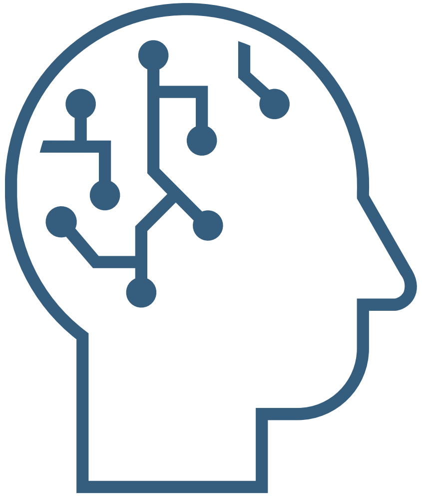
```
0     Married
1      Single
2     Married
3      Single
4      Single
Name: Marital_Status, dtype: object
```

	MAR_ST__Married	MAR_ST__Single	MAR_ST__Unknown
0	1	0	0
1	0	1	0
2	1	0	0
3	0	1	0
4	0	1	0

Original → Processed Data

10127 rows	10127 rows
23 columns	31 columns
0 duplicate or null rows	0 duplicate or null rows

Training the Models



```
# build model #1: logistic regression

model_1 = LogisticRegression()
model_1.fit(train_X, train_y)
y_pred_1 = model_1.predict(test_X)

# build model #2: decision tree

model_2 = DecisionTreeClassifier(random_state=100)
model_2.fit(train_X, train_y)
y_pred_2 = model_2.predict(test_X)

# build model #3: random forest
model_3=RandomForestClassifier(random_state=100)
model_3.fit(train_X, train_y)
y_pred_3 = model_3.predict(test_X)
```

Evaluating Model Results

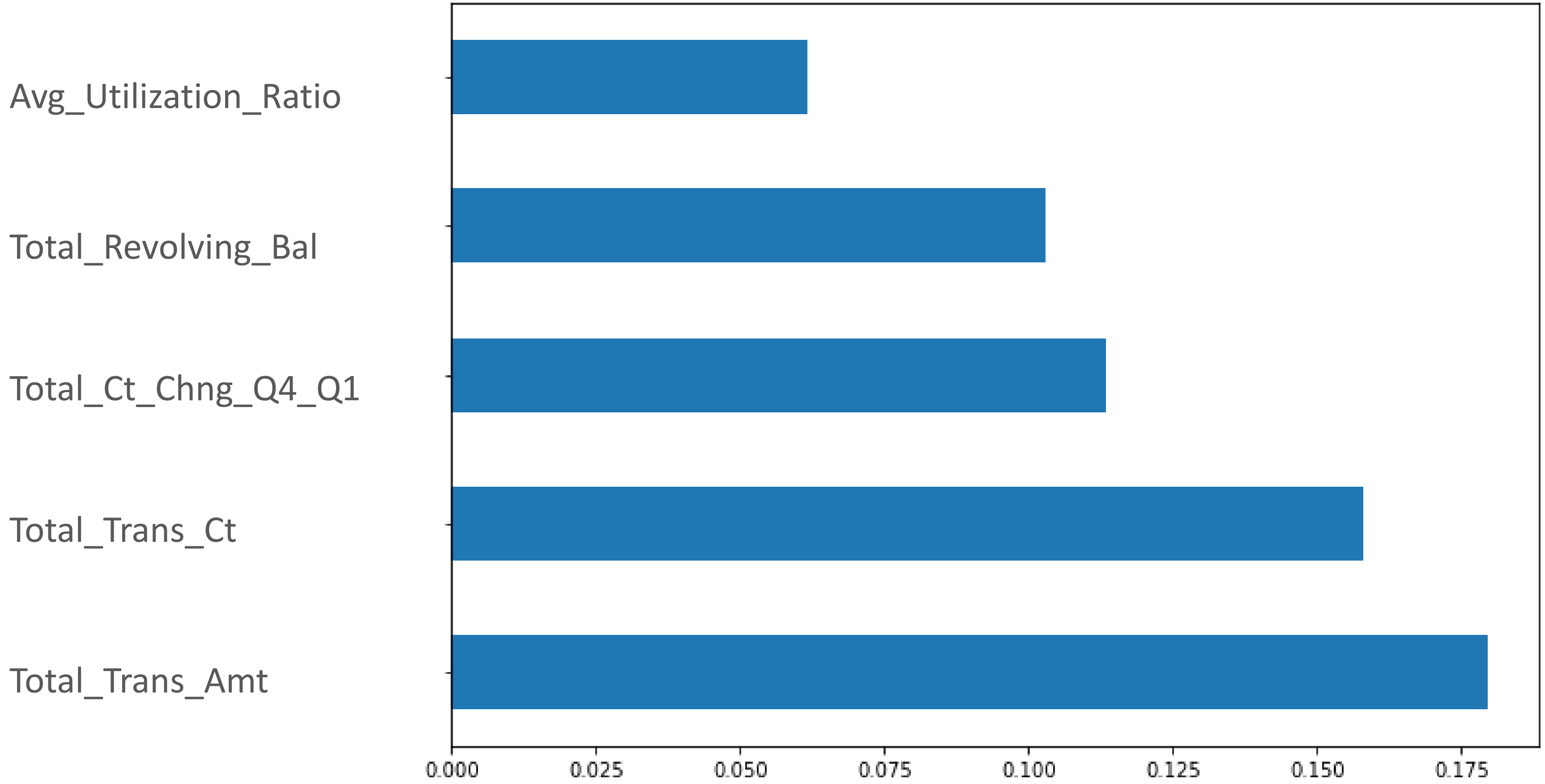
```python
# show accuracy score, confusion matrix, and recall & precision scores

# for model_1
print("\n Evaluation Metrics for model_1 \n")
print("Accuracy Score:", "{:.2%}".format(accuracy_score(test_y, y_pred_1)))
print("Confusion Matrix:\n", confusion_matrix(test_y, y_pred_1))
print("Recall Score:", "{:.2%}".format(recall_score(test_y, y_pred_1)))
print("Precision Score:", "{:.2%}".format(precision_score(test_y, y_pred_1)))

# for model_2
print("\n Evaluation metrics for model_2 \n")
print("Accuracy Score:", "{:.2%}".format(accuracy_score(test_y, y_pred_2)))
print("Confusion Matrix:\n", confusion_matrix(test_y, y_pred_2))
print("Recall Score:", "{:.2%}".format(recall_score(test_y, y_pred_2)))
print("Precision Score:", "{:.2%}".format(precision_score(test_y, y_pred_2)))

# for model_3
print("\n valuation metrics for model_3 \n")
print("Accuracy Score:", "{:.2%}".format(accuracy_score(test_y, y_pred_3)))
print("Confusion Matrix:\n", confusion_matrix(test_y, y_pred_3))
print("Recall Score:", "{:.2%}".format(recall_score(test_y, y_pred_3)))
print("Precision Score:", "{:.2%}".format(precision_score(test_y, y_pred_3)))
```

```
 Evaluation Metrics for model_1

Accuracy Score: 89.88%
Confusion Matrix:
 [[ 199  152]
 [  53 1622]]
Recall Score: 96.84%
Precision Score: 91.43%

 Evaluation metrics for model_2

Accuracy Score: 93.58%
Confusion Matrix:
 [[ 277   74]
 [  56 1619]]
Recall Score: 96.66%
Precision Score: 95.63%

 valuation metrics for model_3

Accuracy Score: 95.56%
Confusion Matrix:
 [[ 282   69]
 [  21 1654]]
Recall Score: 98.75%
Precision Score: 96.00%
```

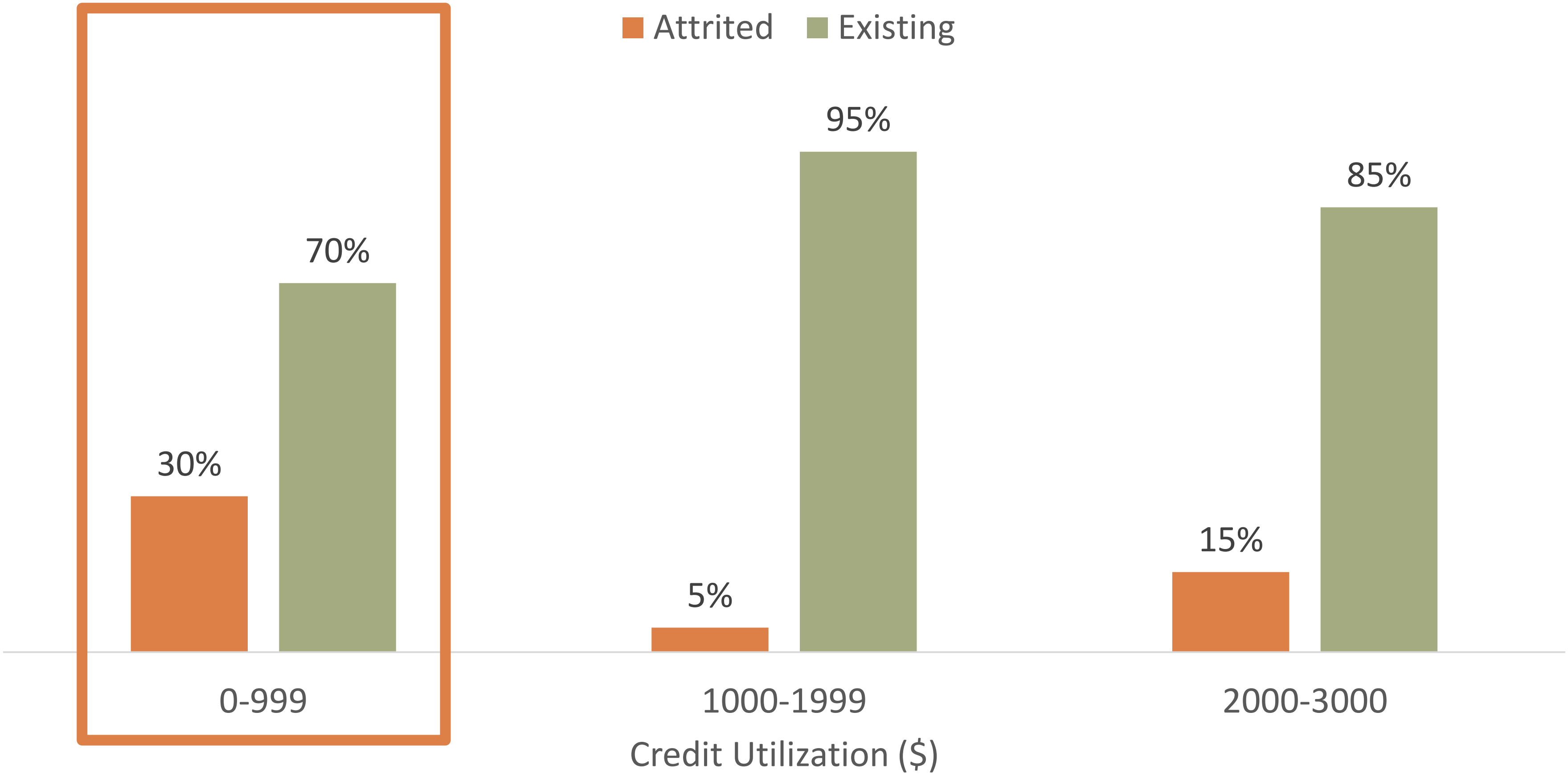
INSIGHTS FROM THE MODEL

Results from statistical model.

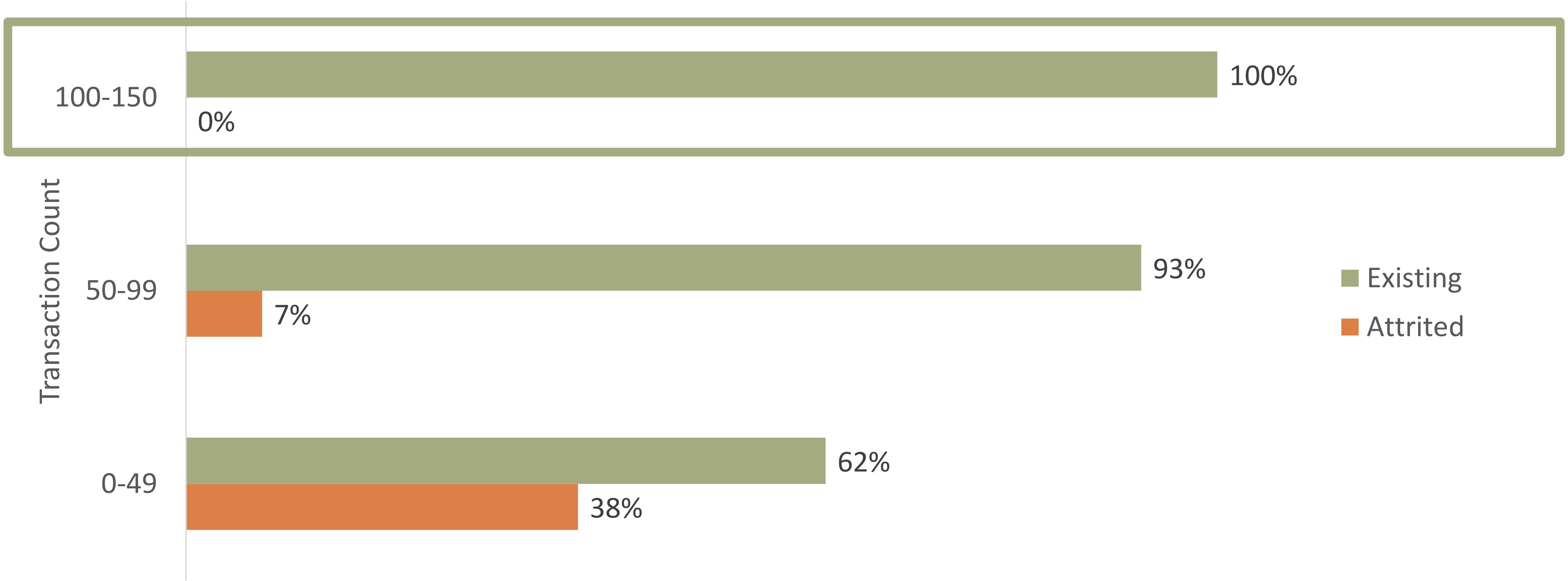
These **5 features** below show the highest level of importance with **Random Forest Regression**.



30 out of 100 customers are likely to churn if their revolving balance is **below $1K**.



There are no **attrited customers** that have made **more than 100 transactions**.



RECOMMENDATIONS

Findings after analyzing the dataset and evaluating the statistical model.

Increase customer engagement and experience by:

1. Offering rewards and incentives.

2. Providing tailored products and services.

3. Developing an omnichannel approach.

REFERENCES

- Kaggle Database https://www.kaggle.com/sakshigoyal7/credit-card-customers
- Winning New Business in Construction By Terry Gillen (2005), p89. Published by Gower Publishing Ltd. ISBN 0566086158. Extracted on 07/2021 from https://www.linkedin.com/pulse/what-cost-customer-acquisition-vs-retention-ian-kingwill/
- Pandas Documentation: https://pandas.pydata.org/docs/
- Numpy Documentation: https://numpy.org/doc/stable/reference/
- SciKit Documentation: https://scikit-learn.org/stable/user_guide.html
- Matplotlib Documentation: https://matplotlib.org/stable/contents.html
- SciPy Documentation: https://docs.scipy.org/doc/scipy/reference/