Sections required in your report:

- Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation and the benefits that your analysis provides to the business or stakeholders of this data.
- Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.
- Brief summary of data exploration and actions taken for data cleaning and feature engineering.
- Summary of training at least three different classifier models, preferably of different nature in explainability and predictability. For example, you can start with a simple logistic regression as a baseline, adding other models or ensemble models. Preferably, all your models use the same training and test splits, or the same cross-validation method.
- A paragraph explaining which of your classifier models you recommend as a final model that best fits your needs in terms of accuracy and explainability.
- Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your classifier model.
- Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model after adding specific data features that may help you achieve a better explanation or a better prediction.

Main Objective

Predicting if a customer will churn, benifits are self explanitory.

In [1]:
```
import pandas as pd
import numpy as np

import seaborn as sns
from matplotlib import pyplot as plt
%matplotlib inline
```

Data Description:

Exploritory Data Analysis

In [2]:
```
df = pd.read_csv('Telco-Customer-Churn.csv')
```

In [3]:
```
len(df)
```

Out[3]:
```
7043
```

In [4]:
```
df.head()
```

Out[4]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Internet!
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fib

5 rows × 21 columns

In [5]:

```
df.head().T
```

Out[5]:

	0	1	2	3	4
customerID	7590-VHVEG	5575-GNVDE	3668-QPYBK	7795-CFOCW	9237-HQITU
gender	Female	Male	Male	Male	Female
SeniorCitizen	0	0	0	0	0
Partner	Yes	No	No	No	No
Dependents	No	No	No	No	No
tenure	1	34	2	45	2
PhoneService	No	Yes	Yes	No	Yes
MultipleLines	No phone service	No	No	No phone service	No
InternetService	DSL	DSL	DSL	DSL	Fiber optic
OnlineSecurity	No	Yes	Yes	Yes	No
OnlineBackup	Yes	No	Yes	No	No
DeviceProtection	No	Yes	No	Yes	No
TechSupport	No	No	No	Yes	No
StreamingTV	No	No	No	No	No
StreamingMovies	No	No	No	No	No
Contract	Month-to-month	One year	Month-to-month	One year	Month-to-month
PaperlessBilling	Yes	No	Yes	No	Yes
PaymentMethod	Electronic check	Mailed check	Mailed check	Bank transfer (automatic)	Electronic check
MonthlyCharges	29.85	56.95	53.85	42.3	70.7
TotalCharges	29.85	1889.5	108.15	1840.75	151.65
Churn	No	No	Yes	No	Yes

In [6]:

```
df.dtypes
```

```
customerID          object
```

Out[6]:
```
gender                object
SeniorCitizen          int64
Partner               object
Dependents            object
tenure                 int64
PhoneService          object
MultipleLines         object
InternetService       object
OnlineSecurity        object
OnlineBackup          object
DeviceProtection      object
TechSupport           object
StreamingTV           object
StreamingMovies       object
Contract              object
PaperlessBilling      object
PaymentMethod         object
MonthlyCharges       float64
TotalCharges          object
Churn                 object
dtype: object
```

TotalCharges isn't correctly identified as a numeric type. we can force thi scolumn to be numeric by converting it to numbers using a special function in pandas: to_numeric. By default, thi sfunction raises an exception when it sees nonnumeric data (such as spaces), but we can make it skip these cases by specifying the errors='coerce' option. Thi sway pandas will replace all nonnumeric values iwht NaN

then we will set the missing values to zero

In [7]:
```
total_charges = pd.to_numeric(df.TotalCharges, errors='coerce')
df[total_charges.isnull()][['customerID', 'TotalCharges']]
df.TotalCharges = df.TotalCharges.fillna(0)
```

The column names don't follow ther same naming convention. Some of them start with a lower letter, whereas others start with a capital letter, and there are also spaces in the vlaues.

we will make it uniform by lowercasing everything and replacing spaces with underscores.

In [8]:
```
df.columns = df.columns.str.lower().str.replace(' ', '_')
string_columns = list(df.dtypes[df.dtypes == 'object'].index)

for col in string_columns:
    df[col] = df[col].str.lower().str.replace(' ', '_')
```

Target variable is churn.

its categorical with two values 'yes' and 'no'. we need to convert these values for binary classification.

In [9]:
```
df.churn = (df.churn == 'yes').astype(int)
```

In [10]:
```
from sklearn.model_selection import train_test_split
```

the function train_test_split takes a dataframe df and creates two new dataframes: df_train_full and df_test

In [11]:
```
df_train_full, df_test = train_test_split(df,test_size=0.2,
                                          random_state=1)
```

In [12]:
```
df_train_full.head()
```

Out[12]:

	customerid	gender	seniorcitizen	partner	dependents	tenure	phoneservice	multiplelines	intern
1814	5442-pptjy	male	0	yes	yes	12	yes	no	
5946	6261-rcvns	female	0	no	no	42	yes	no	
3881	2176-osjuv	male	0	yes	no	71	yes	yes	
2389	6161-erdgd	male	0	yes	yes	71	yes	yes	
3676	2364-ufrom	male	0	no	no	30	yes	no	

5 rows × 21 columns

we want to split the data into three parts:

- train
- validation
- test

since train_test_split splits the data into only two parts: train and test, we will take the df_train_full dataframe and split it one more time into train and validation.

In [13]:

```
df_train, df_val = train_test_split(df_train_full, test_size=0.33,
                                    random_state=11)

y_train = df_train.churn.values
y_val = df_val.churn.values

del df_train['churn']
del df_val['churn']
```

The dataframes are now prepared and we can use the training dataset for performing initial exploritory data analysis (EDA).

In [14]:

```
df_train_full.isnull().sum()
```

Out[14]:

```
customerid          0
gender              0
seniorcitizen       0
partner             0
dependents          0
tenure              0
phoneservice        0
multiplelines       0
internetservice     0
onlinesecurity      0
onlinebackup        0
deviceprotection    0
techsupport         0
streamingtv         0
streamingmovies     0
contract            0
paperlessbilling    0
paymentmethod       0
monthlycharges      0
totalcharges        0
churn               0
dtype: int64
```

we get all zeros so we know we have no missing values we also need to check the distribution of values in the target variable. We can do that with the value_counts() method.

In [15]:
```
df_train_full.churn.value_counts()
```

Out[15]:
```
0    4113
1    1521
Name: churn, dtype: int64
```

The first column is the value of the target variable, and the second is the count. we can see the majority of the customers didn't churn.

In [16]:
```
1521/5634
```

Out[16]:
```
0.26996805111821087
```

We can also calculate the churn rate thats more convenient with the mean() method.

In [17]:
```
global_mean = df_train_full.churn.mean()
global_mean
```

Out[17]:
```
0.26996805111821087
```

In [18]:
```
categorical = ['gender', 'seniorcitizen', 'partner', 'dependents', 'phoneservice', 'mult
numerical = ['tenure', 'monthlycharges', 'totalcharges']
```

In [19]:
```
df_train_full[categorical].nunique()
```

Out[19]:
```
gender              2
seniorcitizen       2
partner             2
dependents          2
phoneservice        2
multiplelines       3
internetservice     3
onlinesecurity      3
onlinebackup        3
deviceprotection    3
techsupport         3
streamingtv         3
streamingmovies     3
contract            3
paperlessbilling    2
paymentmethod       4
dtype: int64
```

In [20]:
```
female_mean = df_train_full[df_train_full.gender == 'female'].churn.mean()
```

In [21]:
```
male_mean = df_train_full[df_train_full.gender == 'male'].churn.mean()
```

In [22]:
```
female_mean
```

Out[22]:
```
0.27682403433476394
```

In [23]:
```
male_mean
```

Out[23]:
```
0.2632135306553911
```

In [24]:
```
female_mean - male_mean
```

Out[24]: 0.013610503679372832

In [25]:
```
partner_yes = df_train_full[df_train_full.partner == 'yes'].churn.mean()
print('partner == yes:', round(partner_yes, 3))

partner_no = df_train_full[df_train_full.partner == 'no'].churn.mean()
print('partner == no:', round(partner_no, 3))
```

```
partner == yes: 0.205
partner == no: 0.33
```

risk ratio

risk = group rate / global rate

ex. for gender == female risk = 27.2% / 27% = 1.02

To check all the values a variable has and compute the churn rate for each of these valeus we will use a rough pandas translation of the below sql

SELECT gender, AVG(churn), AVG(churn) - global_churn, AVG(churn) / global_churn FROM data GROUP BY gender

In [26]:
```
global_mean = df_train_full.churn.mean()

df_group = df_train_full.groupby(by='gender').churn.agg(['mean'])
df_group['diff'] = df_group['mean']- global_mean
df_group['risk'] = df_group['mean'] / global_mean

df_group
```

Out[26]:

gender	mean	diff	risk
female	0.276824	0.006856	1.025396
male	0.263214	-0.006755	0.974980

now do the same thing for categorical variables.

In [27]:
```
from IPython.display import display

for col in categorical:
    df_group = df_train_full.groupby(by=col).churn.agg(['mean'])
    df_group['diff'] = df_group['mean'] - global_mean
    df_group['rate'] = df_group['mean'] / global_mean
    display(df_group)
```

gender	mean	diff	rate
female	0.276824	0.006856	1.025396
male	0.263214	-0.006755	0.974980

seniorcitizen	mean	diff	rate
0	0.242270	-0.027698	0.897403

	mean	diff	rate
1	0.413377	0.143409	1.531208

partner	mean	diff	rate
no	0.329809	0.059841	1.221659
yes	0.205033	-0.064935	0.759472

dependents	mean	diff	rate
no	0.313760	0.043792	1.162212
yes	0.165666	-0.104302	0.613651

phoneservice	mean	diff	rate
no	0.241316	-0.028652	0.893870
yes	0.273049	0.003081	1.011412

multiplelines	mean	diff	rate
no	0.257407	-0.012561	0.953474
no_phone_service	0.241316	-0.028652	0.893870
yes	0.290742	0.020773	1.076948

internetservice	mean	diff	rate
dsl	0.192347	-0.077621	0.712482
fiber_optic	0.425171	0.155203	1.574895
no	0.077805	-0.192163	0.288201

onlinesecurity	mean	diff	rate
no	0.420921	0.150953	1.559152
no_internet_service	0.077805	-0.192163	0.288201
yes	0.153226	-0.116742	0.567570

onlinebackup	mean	diff	rate
no	0.404323	0.134355	1.497672
no_internet_service	0.077805	-0.192163	0.288201
yes	0.217232	-0.052736	0.804660

deviceprotection	mean	diff	rate

	mean	diff	rate
no	0.395875	0.125907	1.466379
no_internet_service	0.077805	-0.192163	0.288201
yes	0.230412	-0.039556	0.853480

techsupport	mean	diff	rate
no	0.418914	0.148946	1.551717
no_internet_service	0.077805	-0.192163	0.288201
yes	0.159926	-0.110042	0.592390

streamingtv	mean	diff	rate
no	0.342832	0.072864	1.269897
no_internet_service	0.077805	-0.192163	0.288201
yes	0.302723	0.032755	1.121328

streamingmovies	mean	diff	rate
no	0.338906	0.068938	1.255358
no_internet_service	0.077805	-0.192163	0.288201
yes	0.307273	0.037305	1.138182

contract	mean	diff	rate
month-to-month	0.431701	0.161733	1.599082
one_year	0.120573	-0.149395	0.446621
two_year	0.028274	-0.241694	0.104730

paperlessbilling	mean	diff	rate
no	0.172071	-0.097897	0.637375
yes	0.338151	0.068183	1.252560

paymentmethod	mean	diff	rate
bank_transfer_(automatic)	0.168171	-0.101797	0.622928
credit_card_(automatic)	0.164339	-0.105630	0.608733
electronic_check	0.455890	0.185922	1.688682
mailed_check	0.193870	-0.076098	0.718121

In [28]:

```
from sklearn.metrics import mutual_info_score

def calculate_mi(series):
    return mutual_info_score(series, df_train_full.churn)
```

```
df_mi = df_train_full[categorical].apply(calculate_mi)
df_mi = df_mi.sort_values(ascending=False).to_frame(name='MI')
df_mi
```

Out[28]:

	MI
contract	0.098320
onlinesecurity	0.063085
techsupport	0.061032
internetservice	0.055868
onlinebackup	0.046923
deviceprotection	0.043453
paymentmethod	0.043210
streamingtv	0.031853
streamingmovies	0.031581
paperlessbilling	0.017589
dependents	0.012346
partner	0.009968
seniorcitizen	0.009410
multiplelines	0.000857
phoneservice	0.000229
gender	0.000117

In [29]:
```
df_train_full[numerical].corrwith(df_train_full.churn)
```

Out[29]:
```
tenure           -0.351885
monthlycharges    0.196805
dtype: float64
```

In [30]:
```
train_dict = df_train[categorical + numerical].to_dict(orient='records')
```

In [31]:
```
from sklearn.feature_extraction import DictVectorizer

dv = DictVectorizer(sparse=False)
dv.fit(train_dict)
```

Out[31]:
```
DictVectorizer(sparse=False)
```

In [32]:
```
X_train = dv.transform(train_dict)
```

In [33]:
```
X_train[0]
```

Out[33]:
```
array([0., 0., 1., ..., 0., 0., 0.])
```

In [34]:
```
dv.get_feature_names_out()
```

Out[34]:
```
array(['contract=month-to-month', 'contract=one_year',
       'contract=two_year', ..., 'totalcharges=999.45',
       'totalcharges=999.9', 'totalcharges=_'], dtype=object)
```

In [35]:
```
def logistic_regression(xi):
    score = bias
```

```
    for j in range(n):
        score = score +xi[j] * w[j]
    prob = sigmoid(score)
```

In [36]:
```
import math
def sigmoid(score):
    return 1 / (1 + math.exp(-score))
```

In [37]:
```
from sklearn.linear_model import LogisticRegression
```

In [38]:
```
model = LogisticRegression(solver='liblinear', random_state=1)
model.fit(X_train, y_train)
```

Out[38]:
```
LogisticRegression(random_state=1, solver='liblinear')
```

In [39]:
```
val_dict = df_val[categorical + numerical].to_dict(orient='records')
X_val = dv.transform(val_dict)
```

In [40]:
```
y_pred = model.predict_proba(X_val)
```

In [41]:
```
y_pred = model.predict_proba(X_val)[:, 1]
```

In [42]:
```
churn = y_pred >= 0.5
```

In [43]:
```
(y_val == churn).mean()
```

Out[43]:
```
0.8064516129032258
```

To see which feature is associated with each weight, use the get_feature_names method of the DictVectorizer. Then zip the feature names together with the coefficients before looking at them

In [44]:
```
dict(zip(dv.get_feature_names_out(), model.coef_[0].round(3)))
```

Out[44]:
```
{'contract=month-to-month': 0.59,
 'contract=one_year': 0.006,
 'contract=two_year': -0.735,
 'dependents=no': -0.021,
 'dependents=yes': -0.119,
 'deviceprotection=no': 0.077,
 'deviceprotection=no_internet_service': -0.18,
 'deviceprotection=yes': -0.037,
 'gender=female': -0.034,
 'gender=male': -0.105,
 'internetservice=dsl': -0.43,
 'internetservice=fiber_optic': 0.471,
 'internetservice=no': -0.18,
 'monthlycharges': -0.001,
 'multiplelines=no': -0.191,
 'multiplelines=no_phone_service': 0.018,
 'multiplelines=yes': 0.034,
 'onlinebackup=no': 0.103,
 'onlinebackup=no_internet_service': -0.18,
 'onlinebackup=yes': -0.062,
 'onlinesecurity=no': 0.24,
 'onlinesecurity=no_internet_service': -0.18,
 'onlinesecurity=yes': -0.199,
 'paperlessbilling=no': -0.227,
 'paperlessbilling=yes': 0.087,
 'partner=no': -0.055,
 'partner=yes': -0.084,
 'paymentmethod=bank_transfer_(automatic)': -0.046,
```

'paymentmethod=credit_card_(automatic)': -0.148,
'paymentmethod=electronic_check': 0.19,
'paymentmethod=mailed_check': -0.136,
'phoneservice=no': 0.018,
'phoneservice=yes': -0.157,
'seniorcitizen': 0.278,
'streamingmovies=no': -0.133,
'streamingmovies=no_internet_service': -0.18,
'streamingmovies=yes': 0.173,
'streamingtv=no': -0.146,
'streamingtv=no_internet_service': -0.18,
'streamingtv=yes': 0.187,
'techsupport=no': 0.17,
'techsupport=no_internet_service': -0.18,
'techsupport=yes': -0.129,
'tenure': -0.034,
'totalcharges=100.25': 0.191,
'totalcharges=100.4': 0.356,
'totalcharges=100.9': -0.105,
'totalcharges=1001.5': -0.015,
'totalcharges=1004.5': -0.02,
'totalcharges=1004.75': -0.18,
'totalcharges=1006.9': -0.013,
'totalcharges=1007.8': -0.023,
'totalcharges=1007.9': -0.016,
'totalcharges=1008.55': 0.286,
'totalcharges=1008.7': 0.258,
'totalcharges=1009.25': -0.243,
'totalcharges=101.1': -0.214,
'totalcharges=101.9': -0.143,
'totalcharges=1011.05': -0.254,
'totalcharges=1011.5': -0.222,
'totalcharges=1011.8': -0.564,
'totalcharges=1012.4': -0.015,
'totalcharges=1013.2': -0.011,
'totalcharges=1013.6': -0.032,
'totalcharges=1016.7': -0.009,
'totalcharges=1017.35': -0.537,
'totalcharges=102.45': 0.216,
'totalcharges=102.75': 0.557,
'totalcharges=1020.2': -0.539,
'totalcharges=1021.55': -0.556,
'totalcharges=1021.75': -0.582,
'totalcharges=1022.5': -0.2,
'totalcharges=1022.6': -0.012,
'totalcharges=1022.95': -0.018,
'totalcharges=1023.75': -0.196,
'totalcharges=1023.9': 0.493,
'totalcharges=1024': -0.052,
'totalcharges=1024.65': -0.008,
'totalcharges=1025.15': -0.235,
'totalcharges=1025.95': 0.39,
'totalcharges=1026.35': -0.013,
'totalcharges=1028.75': -0.008,
'totalcharges=1029.35': -0.394,
'totalcharges=1029.8': -0.01,
'totalcharges=103.7': -0.261,
'totalcharges=1032': 0.498,
'totalcharges=1032.05': -0.019,
'totalcharges=1036': -0.012,
'totalcharges=1036.75': 0.239,
'totalcharges=1041.8': -0.013,
'totalcharges=1042.65': -0.008,
'totalcharges=1043.3': 0.377,
'totalcharges=1043.35': 0.479,
'totalcharges=1043.8': -0.371,

'totalcharges=1045.25': -0.186,
'totalcharges=1046.1': 0.173,
'totalcharges=1046.2': -0.01,
'totalcharges=1046.5': 0.202,
'totalcharges=1047.7': 0.237,
'totalcharges=1049.05': -0.213,
'totalcharges=1051.9': -0.008,
'totalcharges=1052.35': 0.968,
'totalcharges=1054.75': -0.007,
'totalcharges=1055.9': -0.081,
'totalcharges=1057.85': -0.007,
'totalcharges=1058.1': -0.141,
'totalcharges=1058.6': 0.316,
'totalcharges=1059.55': 0.222,
'totalcharges=106.2': -0.245,
'totalcharges=106.8': -0.16,
'totalcharges=106.9': -0.033,
'totalcharges=1060.2': 0.353,
'totalcharges=1061.6': -0.012,
'totalcharges=1064.65': 0.244,
'totalcharges=1066.15': -0.037,
'totalcharges=1066.9': -0.613,
'totalcharges=1067.15': 0.706,
'totalcharges=1067.65': -0.008,
'totalcharges=107.05': -0.117,
'totalcharges=107.1': -0.304,
'totalcharges=107.6': -0.042,
'totalcharges=1070.15': -0.024,
'totalcharges=1070.25': -0.008,
'totalcharges=1070.5': -0.007,
'totalcharges=1070.7': -0.135,
'totalcharges=1071.4': -0.193,
'totalcharges=1072': 0.293,
'totalcharges=1072.6': -0.069,
'totalcharges=1073.3': -0.451,
'totalcharges=1077.5': 0.642,
'totalcharges=1078.9': 0.172,
'totalcharges=1079.45': -0.008,
'totalcharges=1079.65': -0.006,
'totalcharges=108.15': 0.596,
'totalcharges=1083.7': -0.272,
'totalcharges=1086.75': -0.007,
'totalcharges=1087.25': -0.024,
'totalcharges=1087.7': -0.009,
'totalcharges=1088.25': -0.011,
'totalcharges=109.2': -0.121,
'totalcharges=109.25': -0.276,
'totalcharges=109.5': -0.154,
'totalcharges=109.8': -0.117,
'totalcharges=1090.65': -0.029,
'totalcharges=1093': -0.006,
'totalcharges=1093.1': 0.458,
'totalcharges=1093.2': -0.235,
'totalcharges=1095.3': 0.428,
'totalcharges=1095.65': 0.321,
'totalcharges=1096.25': 0.228,
'totalcharges=1096.65': -0.098,
'totalcharges=1098.85': -0.441,
'totalcharges=110.15': -0.114,
'totalcharges=1101.85': 0.255,
'totalcharges=1103.25': -0.008,
'totalcharges=1107.2': -0.009,
'totalcharges=1107.25': 0.355,
'totalcharges=1108': -0.276,
'totalcharges=1108.6': -0.426,
'totalcharges=111.05': -0.292,

'totalcharges=1110.05': -0.007,
'totalcharges=1110.35': -0.009,
'totalcharges=1110.5': -0.023,
'totalcharges=1111.2': -0.222,
'totalcharges=1112.3': -0.34,
'totalcharges=1114.55': -0.346,
'totalcharges=1114.85': -0.042,
'totalcharges=1115.15': 0.435,
'totalcharges=1119.35': -0.141,
'totalcharges=112.3': 0.716,
'totalcharges=112.75': -0.121,
'totalcharges=1120.3': 0.183,
'totalcharges=1120.95': 0.365,
'totalcharges=1121.05': -0.013,
'totalcharges=1122.4': -0.552,
'totalcharges=1123.15': -0.006,
'totalcharges=1123.65': -0.008,
'totalcharges=1125.6': -0.007,
'totalcharges=1126.75': -0.014,
'totalcharges=1129.35': -0.461,
'totalcharges=1129.75': 0.184,
'totalcharges=113.1': -0.312,
'totalcharges=113.35': 0.587,
'totalcharges=113.85': 0.758,
'totalcharges=1130.85': 0.348,
'totalcharges=1131.2': 0.153,
'totalcharges=1132.35': 0.434,
'totalcharges=1132.75': 0.276,
'totalcharges=1133.7': -0.027,
'totalcharges=1134.25': -0.01,
'totalcharges=1135.7': 0.217,
'totalcharges=1137.05': -0.258,
'totalcharges=1138.8': -0.017,
'totalcharges=114.1': 0.464,
'totalcharges=1144.5': 0.328,
'totalcharges=1144.6': -0.207,
'totalcharges=1145.35': 0.195,
'totalcharges=1145.7': 0.286,
'totalcharges=1146.05': -0.007,
'totalcharges=1146.65': 0.296,
'totalcharges=1149.65': -0.007,
'totalcharges=115.1': 0.509,
'totalcharges=1151.05': -0.099,
'totalcharges=1151.55': -0.254,
'totalcharges=1155.6': -0.217,
'totalcharges=1156.35': -0.2,
'totalcharges=1157.05': 0.574,
'totalcharges=1158.85': -0.008,
'totalcharges=116.85': -0.038,
'totalcharges=1160.45': -0.01,
'totalcharges=1161.75': -0.063,
'totalcharges=1164.05': -0.315,
'totalcharges=1164.3': 0.221,
'totalcharges=1165.6': -0.371,
'totalcharges=1165.9': -0.023,
'totalcharges=1166.7': -0.249,
'totalcharges=1169.35': 0.347,
'totalcharges=117.8': -0.074,
'totalcharges=1171.3': -0.016,
'totalcharges=1171.5': -0.024,
'totalcharges=1173.35': -0.01,
'totalcharges=1174.35': 0.425,
'totalcharges=1174.8': -0.03,
'totalcharges=1175.85': -0.168,
'totalcharges=1177.05': -0.007,
'totalcharges=1177.95': -0.006,

'totalcharges=1178.25': 0.383,
'totalcharges=1178.4': 0.282,
'totalcharges=118.3': 0.417,
'totalcharges=118.5': 0.367,
'totalcharges=1180.95': 0.289,
'totalcharges=1182.55': 0.924,
'totalcharges=1183.05': -0.612,
'totalcharges=1183.8': -0.255,
'totalcharges=1184': -0.418,
'totalcharges=1188.25': -0.006,
'totalcharges=1189.9': -0.006,
'totalcharges=119.75': 0.293,
'totalcharges=1191.4': -0.006,
'totalcharges=1192.3': -0.009,
'totalcharges=1192.7': -0.005,
'totalcharges=1193.05': 0.789,
'totalcharges=1195.25': -0.011,
'totalcharges=1195.95': -0.014,
'totalcharges=1199.4': -0.191,
'totalcharges=120.25': -0.574,
'totalcharges=1200.15': 0.162,
'totalcharges=1201.65': -0.157,
'totalcharges=1203.9': -0.581,
'totalcharges=1203.95': -0.027,
'totalcharges=1205.05': -0.005,
'totalcharges=1207': -0.464,
'totalcharges=1208.15': -0.053,
'totalcharges=1208.6': -0.005,
'totalcharges=1209.25': -0.005,
'totalcharges=121.25': -0.058,
'totalcharges=1210.3': -0.007,
'totalcharges=1210.4': 0.686,
'totalcharges=1215.45': -0.158,
'totalcharges=1215.65': -0.607,
'totalcharges=1216.35': -0.025,
'totalcharges=1218.25': -0.187,
'totalcharges=1218.55': 0.172,
'totalcharges=1218.65': -0.007,
'totalcharges=122': -0.107,
'totalcharges=122.9': -0.103,
'totalcharges=1221.65': -0.472,
'totalcharges=1222.05': 0.399,
'totalcharges=1222.25': -0.012,
'totalcharges=1222.65': -0.006,
'totalcharges=1225.65': -0.005,
'totalcharges=123.65': 0.416,
'totalcharges=1230.25': 0.362,
'totalcharges=1233.4': 0.327,
'totalcharges=1234.8': -0.008,
'totalcharges=1235.55': -0.357,
'totalcharges=1236.15': 0.489,
'totalcharges=1237.3': -0.136,
'totalcharges=124.4': -0.07,
'totalcharges=124.45': -0.038,
'totalcharges=1240.15': -0.005,
'totalcharges=1242.2': -0.45,
'totalcharges=1244.8': -0.011,
'totalcharges=1245.05': -0.011,
'totalcharges=1248.9': -0.416,
'totalcharges=1249.25': -0.195,
'totalcharges=125.95': -0.298,
'totalcharges=1252.85': -0.007,
'totalcharges=1253.5': -0.159,
'totalcharges=1253.9': 0.222,
'totalcharges=1258.3': 0.308,
'totalcharges=1259': 0.314,

'totalcharges=1259.35': -0.362,
'totalcharges=126.05': 0.762,
'totalcharges=1261.7': -0.025,
'totalcharges=1263.9': -0.005,
'totalcharges=1264.2': -0.277,
'totalcharges=1265.65': -0.228,
'totalcharges=1266.1': 0.281,
'totalcharges=1266.35': -0.088,
'totalcharges=1266.4': -0.031,
'totalcharges=1267': -0.014,
'totalcharges=1267.95': -0.403,
'totalcharges=1268.85': -0.006,
'totalcharges=127.1': 0.409,
'totalcharges=1270.55': -0.008,
'totalcharges=1271.8': -0.1,
'totalcharges=1274.45': -0.104,
'totalcharges=1275.6': -0.013,
'totalcharges=1275.7': -0.193,
'totalcharges=1277.75': -0.005,
'totalcharges=128.6': -0.068,
'totalcharges=1281': 0.202,
'totalcharges=1282.85': -0.063,
'totalcharges=1284.2': 0.18,
'totalcharges=1285.05': -0.628,
'totalcharges=1288.3': -0.491,
'totalcharges=1288.75': -0.209,
'totalcharges=129.55': -0.119,
'totalcharges=129.6': -0.382,
'totalcharges=1291.35': -0.183,
'totalcharges=1292.6': -0.005,
'totalcharges=1295.4': -0.153,
'totalcharges=1296.15': -0.006,
'totalcharges=1297.35': -0.005,
'totalcharges=1297.8': -0.174,
'totalcharges=1298.7': -0.008,
'totalcharges=130.5': -0.122,
'totalcharges=130.55': -0.142,
'totalcharges=1301': 0.292,
'totalcharges=1301.1': -0.004,
'totalcharges=1301.7': 0.281,
'totalcharges=1303.5': -0.005,
'totalcharges=1304.8': -0.004,
'totalcharges=1304.85': -0.108,
'totalcharges=1305.95': -0.005,
'totalcharges=1306.3': -0.04,
'totalcharges=1308.1': 0.199,
'totalcharges=1309': -0.21,
'totalcharges=131.05': 0.272,
'totalcharges=131.65': 0.201,
'totalcharges=1311.6': -0.005,
'totalcharges=1311.75': -0.005,
'totalcharges=1312.45': -0.144,
'totalcharges=1313.55': 0.364,
'totalcharges=1316.9': -0.072,
'totalcharges=1319.85': -0.116,
'totalcharges=132.4': -0.392,
'totalcharges=1322.55': -0.385,
'totalcharges=1323.7': -0.145,
'totalcharges=1326.35': -0.004,
'totalcharges=1327.85': -0.099,
'totalcharges=1329.2': 0.343,
'totalcharges=1331.05': -0.009,
'totalcharges=1332.4': -0.408,
'totalcharges=1336.1': -0.008,
'totalcharges=1336.15': -0.131,
'totalcharges=1336.35': -0.014,

'totalcharges=1336.9': -0.421,
'totalcharges=1337.45': -0.605,
'totalcharges=1337.5': -0.005,
'totalcharges=1339.8': -0.228,
'totalcharges=134.05': -0.676,
'totalcharges=134.35': 0.426,
'totalcharges=134.5': 0.555,
'totalcharges=1340.1': -0.51,
'totalcharges=1344.5': -0.006,
'totalcharges=1345.55': 0.24,
'totalcharges=1345.85': -0.009,
'totalcharges=1346.9': -0.115,
'totalcharges=1348.95': -0.367,
'totalcharges=135.75': -0.695,
'totalcharges=1355.45': -0.073,
'totalcharges=1356.3': -0.006,
'totalcharges=1357.1': -0.005,
'totalcharges=1362.85': -0.125,
'totalcharges=1364.3': 0.188,
'totalcharges=1367.75': -0.166,
'totalcharges=137.1': -0.467,
'totalcharges=137.25': -0.325,
'totalcharges=137.6': 0.532,
'totalcharges=137.85': 0.449,
'totalcharges=1370.35': -0.245,
'totalcharges=1372.45': -0.004,
'totalcharges=1372.9': -0.007,
'totalcharges=1373.05': -0.352,
'totalcharges=1374.2': -0.004,
'totalcharges=1374.9': -0.32,
'totalcharges=1375.4': -0.005,
'totalcharges=1375.6': -0.178,
'totalcharges=1378.75': -0.004,
'totalcharges=1379.6': -0.004,
'totalcharges=138.85': -0.102,
'totalcharges=1380.1': 0.749,
'totalcharges=1380.4': -0.384,
'totalcharges=1381.2': -0.098,
'totalcharges=1381.8': -0.334,
'totalcharges=1385.85': -0.011,
'totalcharges=1386.8': -0.076,
'totalcharges=1387': -0.376,
'totalcharges=1387.45': -0.006,
'totalcharges=1388.45': -0.005,
'totalcharges=139.05': 0.201,
'totalcharges=139.25': -0.517,
'totalcharges=139.4': 0.322,
'totalcharges=139.75': -0.102,
'totalcharges=1391.15': -0.561,
'totalcharges=1391.65': -0.064,
'totalcharges=1393.6': -0.204,
'totalcharges=1396': -0.008,
'totalcharges=1396.9': -0.005,
'totalcharges=1397.65': -0.005,
'totalcharges=1398.25': -0.22,
'totalcharges=140.4': -0.464,
'totalcharges=140.7': 0.368,
'totalcharges=1400.55': 0.647,
'totalcharges=1400.85': 0.364,
'totalcharges=1401.15': -0.004,
'totalcharges=1402.25': -0.008,
'totalcharges=1403.1': -0.011,
'totalcharges=1404.65': -0.27,
'totalcharges=1406.9': -0.17,
'totalcharges=141.1': -0.111,
'totalcharges=141.45': 0.247,

'totalcharges=1410.25': -0.26,
'totalcharges=1411.2': -0.006,
'totalcharges=1411.35': -0.11,
'totalcharges=1411.9': -0.006,
'totalcharges=1412.4': -0.054,
'totalcharges=1414.2': -0.041,
'totalcharges=1414.65': -0.005,
'totalcharges=1415.55': 0.215,
'totalcharges=1416.2': -0.202,
'totalcharges=1416.5': -0.004,
'totalcharges=1416.75': -0.008,
'totalcharges=1419.4': -0.006,
'totalcharges=142.35': 0.201,
'totalcharges=1421.9': -0.004,
'totalcharges=1422.1': 0.255,
'totalcharges=1423.15': -0.219,
'totalcharges=1423.35': -0.051,
'totalcharges=1424.2': -0.011,
'totalcharges=1424.4': -0.162,
'totalcharges=1424.5': 0.452,
'totalcharges=1424.6': -0.009,
'totalcharges=1424.9': -0.005,
'totalcharges=1424.95': 0.264,
'totalcharges=1425.45': 0.619,
'totalcharges=1426.4': 0.379,
'totalcharges=1427.55': -0.119,
'totalcharges=143.35': 0.21,
'totalcharges=1431.65': -0.021,
'totalcharges=1432.55': -0.005,
'totalcharges=1434.1': -0.004,
'totalcharges=1438.05': -0.321,
'totalcharges=144': -0.566,
'totalcharges=144.15': 0.196,
'totalcharges=144.35': -0.1,
'totalcharges=144.55': 0.205,
'totalcharges=144.95': -0.117,
'totalcharges=1441.65': -0.005,
'totalcharges=1441.95': -0.21,
'totalcharges=1442.2': -0.482,
'totalcharges=1442.6': -0.234,
'totalcharges=1443.65': -0.562,
'totalcharges=1444.05': -0.013,
'totalcharges=1444.65': 0.195,
'totalcharges=1445.2': -0.009,
'totalcharges=1447.9': -0.007,
'totalcharges=1448.8': 0.805,
'totalcharges=145.15': -0.239,
'totalcharges=1451.1': -0.423,
'totalcharges=1451.9': 0.338,
'totalcharges=1453.1': -0.15,
'totalcharges=1454.25': -0.077,
'totalcharges=1457.25': 0.261,
'totalcharges=1458.1': -0.004,
'totalcharges=1459.35': -0.153,
'totalcharges=146.05': 0.195,
'totalcharges=146.4': -0.189,
'totalcharges=146.6': -0.042,
'totalcharges=146.9': 0.364,
'totalcharges=1460.85': -0.004,
'totalcharges=1461.45': -0.086,
'totalcharges=1462.05': -0.008,
'totalcharges=1462.6': -0.165,
'totalcharges=1463.45': 0.385,
'totalcharges=1465.75': 0.284,
'totalcharges=1468.75': -0.005,
'totalcharges=147.15': 0.538,

'totalcharges=147.55': -0.141,
'totalcharges=147.75': -0.273,
'totalcharges=147.8': -0.371,
'totalcharges=1470.05': -0.485,
'totalcharges=1470.95': 0.384,
'totalcharges=1471.75': 0.286,
'totalcharges=1477.65': -0.063,
'totalcharges=1478.85': -0.24,
'totalcharges=1483.25': 0.304,
'totalcharges=1489.3': -0.078,
'totalcharges=149.05': -0.429,
'totalcharges=149.55': 0.188,
'totalcharges=1490.4': -0.368,
'totalcharges=1490.95': 0.408,
'totalcharges=1492.1': -0.004,
'totalcharges=1493.1': -0.011,
'totalcharges=1493.2': -0.452,
'totalcharges=1493.55': -0.32,
'totalcharges=1497.05': -0.187,
'totalcharges=1497.9': -0.021,
'totalcharges=1498.2': 0.628,
'totalcharges=1498.55': -0.147,
'totalcharges=1498.65': -0.034,
'totalcharges=1498.85': -0.009,
'totalcharges=150.35': -0.12,
'totalcharges=150.6': -0.117,
'totalcharges=150.75': -0.106,
'totalcharges=150.85': -0.27,
'totalcharges=1500.5': -0.324,
'totalcharges=1500.95': -0.004,
'totalcharges=1501.75': -0.01,
'totalcharges=1502.25': -0.004,
'totalcharges=1504.05': 0.216,
'totalcharges=1505.05': -0.008,
'totalcharges=1505.9': -0.017,
'totalcharges=1506.4': -0.014,
'totalcharges=1507': -0.102,
'totalcharges=151.3': 0.424,
'totalcharges=151.75': -0.527,
'totalcharges=1510.3': -0.111,
'totalcharges=1513.6': -0.011,
'totalcharges=1514.85': -0.008,
'totalcharges=1515.1': 0.222,
'totalcharges=1516.6': -0.076,
'totalcharges=1517.5': -0.119,
'totalcharges=152.3': -0.273,
'totalcharges=152.45': 0.326,
'totalcharges=152.6': -0.125,
'totalcharges=152.95': -0.179,
'totalcharges=1520.1': -0.057,
'totalcharges=1520.9': 0.337,
'totalcharges=1521.2': -0.006,
'totalcharges=1522.7': -0.362,
'totalcharges=1524.85': -0.038,
'totalcharges=1525.35': -0.077,
'totalcharges=1527.35': 0.329,
'totalcharges=1529.2': 0.156,
'totalcharges=1529.65': -0.004,
'totalcharges=153.3': -0.039,
'totalcharges=1531.4': 0.183,
'totalcharges=1532.45': 0.304,
'totalcharges=1533.8': -0.056,
'totalcharges=1534.05': -0.007,
'totalcharges=1534.75': 0.218,
'totalcharges=1536.75': -0.138,
'totalcharges=1537.85': -0.055,

'totalcharges=1537.9': -0.456,
'totalcharges=1539.8': -0.554,
'totalcharges=154.55': -0.249,
'totalcharges=154.65': 0.634,
'totalcharges=154.85': 0.248,
'totalcharges=1540.05': -0.494,
'totalcharges=1540.2': -0.177,
'totalcharges=1540.35': -0.406,
'totalcharges=1546.3': -0.053,
'totalcharges=155.35': -0.246,
'totalcharges=155.8': 0.334,
'totalcharges=155.9': -0.117,
'totalcharges=1551.6': -0.009,
'totalcharges=1553.2': -0.109,
'totalcharges=1554.9': -0.008,
'totalcharges=1556.85': -0.321,
'totalcharges=1559.25': -0.528,
'totalcharges=1559.45': -0.117,
'totalcharges=156.25': 0.753,
'totalcharges=1563.95': -0.01,
'totalcharges=1566.75': -0.555,
'totalcharges=1567': -0.005,
'totalcharges=157.65': -0.111,
'totalcharges=1573.95': -0.231,
'totalcharges=1574.45': 0.497,
'totalcharges=1574.5': -0.006,
'totalcharges=158.35': -0.105,
'totalcharges=1584.8': -0.007,
'totalcharges=1587.55': 0.182,
'totalcharges=1588.7': -0.15,
'totalcharges=159.15': 0.48,
'totalcharges=159.35': -0.034,
'totalcharges=159.4': 0.193,
'totalcharges=159.45': 0.246,
'totalcharges=1595.5': -0.013,
'totalcharges=1596.6': -0.179,
'totalcharges=160.05': -0.083,
'totalcharges=160.75': 0.261,
'totalcharges=160.85': -0.25,
'totalcharges=1600.25': -0.377,
'totalcharges=1600.95': -0.013,
'totalcharges=1601.5': -0.43,
'totalcharges=1604.5': 0.856,
'totalcharges=1607.2': 0.416,
'totalcharges=161.45': -0.559,
'totalcharges=161.5': 0.197,
'totalcharges=161.65': -0.036,
'totalcharges=1611.65': -0.171,
'totalcharges=1612.75': 0.297,
'totalcharges=1614.05': -0.302,
'totalcharges=1614.7': -0.688,
'totalcharges=1615.1': -0.416,
'totalcharges=1618.2': -0.007,
'totalcharges=162.15': -0.032,
'totalcharges=162.45': 0.22,
'totalcharges=1620.2': -0.006,
'totalcharges=1620.25': -0.081,
'totalcharges=1621.35': -0.116,
'totalcharges=1623.4': -0.058,
'totalcharges=1629.2': -0.006,
'totalcharges=163.7': -0.084,
'totalcharges=1630.4': -0.067,
'totalcharges=1636.95': -0.081,
'totalcharges=1637.3': -0.476,
'totalcharges=1638.7': -0.007,
'totalcharges=164.5': 0.195,

'totalcharges=164.6': -0.11,
'totalcharges=164.85': 0.757,
'totalcharges=1640': -0.435,
'totalcharges=1641.3': -0.446,
'totalcharges=1641.8': -0.006,
'totalcharges=1642.75': -0.314,
'totalcharges=165': 0.396,
'totalcharges=165.35': -0.034,
'totalcharges=165.45': 0.179,
'totalcharges=165.5': -0.03,
'totalcharges=1652.4': -0.345,
'totalcharges=1652.95': -0.131,
'totalcharges=1653.45': -0.236,
'totalcharges=1654.75': -0.006,
'totalcharges=166.3': 0.21,
'totalcharges=1660': 0.287,
'totalcharges=1662.05': -0.049,
'totalcharges=1663.5': -0.005,
'totalcharges=1663.75': -0.179,
'totalcharges=1664.3': -0.008,
'totalcharges=1665.2': -0.187,
'totalcharges=1667.25': -0.262,
'totalcharges=1669.4': -0.433,
'totalcharges=167.2': -0.409,
'totalcharges=167.5': 0.211,
'totalcharges=1671.6': -0.572,
'totalcharges=1672.1': -0.322,
'totalcharges=1672.15': -0.008,
'totalcharges=1672.35': -0.255,
'totalcharges=1673.4': -0.008,
'totalcharges=1673.8': -0.008,
'totalcharges=1677.85': -0.264,
'totalcharges=1678.05': -0.006,
'totalcharges=1679.65': -0.202,
'totalcharges=168.15': 0.465,
'totalcharges=168.2': 0.158,
'totalcharges=168.5': -0.521,
'totalcharges=168.65': -0.142,
'totalcharges=1680.25': -0.008,
'totalcharges=1681.6': -0.007,
'totalcharges=1682.05': -0.154,
'totalcharges=1682.4': -0.31,
'totalcharges=1686.15': -0.005,
'totalcharges=1686.85': -0.324,
'totalcharges=169.45': 0.439,
'totalcharges=1692.6': 0.312,
'totalcharges=1696.2': -0.073,
'totalcharges=1697.7': -0.006,
'totalcharges=170.5': -0.507,
'totalcharges=170.9': -0.106,
'totalcharges=1700.9': -0.051,
'totalcharges=1704.95': 0.536,
'totalcharges=1706.45': -0.435,
'totalcharges=171': -0.082,
'totalcharges=171.15': 0.209,
'totalcharges=1710.15': -0.008,
'totalcharges=1712.7': 0.295,
'totalcharges=1714.95': -0.007,
'totalcharges=1716.45': 0.81,
'totalcharges=1718.95': -0.193,
'totalcharges=1719.15': -0.008,
'totalcharges=172.35': -0.061,
'totalcharges=172.85': 0.237,
'totalcharges=1724.15': -0.006,
'totalcharges=1725.95': -0.277,
'totalcharges=1727.5': 0.268,

'totalcharges=1729.35': -0.351,
'totalcharges=173': -0.132,
'totalcharges=173.15': -0.072,
'totalcharges=1730.65': -0.006,
'totalcharges=1732.6': -0.346,
'totalcharges=1734.5': -0.046,
'totalcharges=1734.65': -0.004,
'totalcharges=1738.9': -0.005,
'totalcharges=174.2': -0.385,
'totalcharges=174.3': 0.24,
'totalcharges=174.65': -0.036,
'totalcharges=174.7': -0.138,
'totalcharges=174.8': 0.134,
'totalcharges=1742.45': -0.213,
'totalcharges=1742.5': -0.567,
'totalcharges=1742.75': 0.519,
'totalcharges=1747.2': -0.317,
'totalcharges=1748.55': -0.009,
'totalcharges=1748.9': -0.049,
'totalcharges=1750.85': -0.273,
'totalcharges=1752.45': -0.583,
'totalcharges=1752.55': 0.172,
'totalcharges=1752.65': -0.219,
'totalcharges=1753': -0.006,
'totalcharges=1755.35': -0.018,
'totalcharges=1756.6': -0.218,
'totalcharges=1759.4': -0.059,
'totalcharges=176.2': -0.318,
'totalcharges=1766.75': -0.009,
'totalcharges=1767.35': 0.519,
'totalcharges=177.4': -0.259,
'totalcharges=1775.8': 0.242,
'totalcharges=1776': -0.007,
'totalcharges=1776.95': 0.283,
'totalcharges=1777.6': -0.006,
'totalcharges=1777.9': 0.277,
'totalcharges=1778.7': -0.008,
'totalcharges=1779.95': 0.399,
'totalcharges=178.5': 0.778,
'totalcharges=178.7': -0.1,
'totalcharges=1782': -0.064,
'totalcharges=1783.75': -0.289,
'totalcharges=1784.5': -0.005,
'totalcharges=1784.9': -0.006,
'totalcharges=1789.25': 0.359,
'totalcharges=179.35': 0.616,
'totalcharges=1790.15': -0.005,
'totalcharges=1790.35': -0.009,
'totalcharges=1790.6': -0.068,
'totalcharges=1793.25': -0.336,
'totalcharges=1794.8': -0.043,
'totalcharges=1796.55': -0.007,
'totalcharges=1797.1': -0.005,
'totalcharges=1797.75': -0.396,
'totalcharges=1798.9': -0.007,
'totalcharges=18.8': -0.073,
'totalcharges=18.85': 0.683,
'totalcharges=18.9': -0.131,
'totalcharges=1801.1': -0.089,
'totalcharges=1802.15': -0.224,
'totalcharges=1803.7': -0.36,
'totalcharges=1808.7': 0.268,
'totalcharges=181.1': 0.209,
'totalcharges=181.5': 0.176,
'totalcharges=181.6': 0.441,
'totalcharges=181.65': -0.187,

'totalcharges=181.7': 0.113,
'totalcharges=1810.55': -0.005,
'totalcharges=1816.2': -0.01,
'totalcharges=1818.3': -0.167,
'totalcharges=1821.8': -0.007,
'totalcharges=1821.95': -0.505,
'totalcharges=1825.5': -0.2,
'totalcharges=183.15': -0.078,
'totalcharges=183.75': 0.256,
'totalcharges=1830.05': -0.006,
'totalcharges=1830.1': -0.133,
'totalcharges=1832.4': -0.157,
'totalcharges=184.15': -0.511,
'totalcharges=184.4': -0.064,
'totalcharges=1840.75': -0.041,
'totalcharges=1842.7': -0.018,
'totalcharges=1842.8': 0.197,
'totalcharges=1845.9': 0.42,
'totalcharges=1849.95': -0.521,
'totalcharges=185.2': -0.087,
'totalcharges=185.55': 0.215,
'totalcharges=185.6': 0.729,
'totalcharges=1851.45': -0.004,
'totalcharges=1852.85': 0.358,
'totalcharges=1855.65': 0.398,
'totalcharges=1856.4': -0.012,
'totalcharges=1857.25': -0.417,
'totalcharges=1857.3': -0.074,
'totalcharges=1859.1': -0.004,
'totalcharges=1859.2': -0.005,
'totalcharges=1859.5': -0.467,
'totalcharges=186.15': -0.111,
'totalcharges=186.3': -0.368,
'totalcharges=1862.9': -0.452,
'totalcharges=1863.8': 0.859,
'totalcharges=1864.65': -0.065,
'totalcharges=1866.45': -0.005,
'totalcharges=1867.6': -0.159,
'totalcharges=1868.4': -0.452,
'totalcharges=187.45': 0.136,
'totalcharges=1871.15': -0.365,
'totalcharges=1871.85': -0.005,
'totalcharges=1874.3': -0.005,
'totalcharges=1875.25': -0.019,
'totalcharges=188.7': 0.482,
'totalcharges=1880.85': 0.746,
'totalcharges=1884.65': -0.275,
'totalcharges=1885.15': -0.05,
'totalcharges=1886.4': -0.088,
'totalcharges=1887': -0.006,
'totalcharges=1889.5': -0.049,
'totalcharges=189.45': -0.065,
'totalcharges=189.95': -0.348,
'totalcharges=1893.95': 0.318,
'totalcharges=19': -0.159,
'totalcharges=19.05': -0.131,
'totalcharges=19.1': -0.244,
'totalcharges=19.15': -0.138,
'totalcharges=19.2': -0.269,
'totalcharges=19.25': -0.309,
'totalcharges=19.3': 0.738,
'totalcharges=19.4': -0.28,
'totalcharges=19.45': 0.66,
'totalcharges=19.5': -0.281,
'totalcharges=19.55': -0.344,
'totalcharges=19.6': 1.137,

'totalcharges=19.65': 1.013,
'totalcharges=19.7': -0.118,
'totalcharges=19.75': 0.013,
'totalcharges=19.85': 0.624,
'totalcharges=19.9': 0.936,
'totalcharges=19.95': 0.982,
'totalcharges=190.05': 0.418,
'totalcharges=190.1': -0.364,
'totalcharges=190.5': -0.452,
'totalcharges=1900.25': 0.57,
'totalcharges=1901': -0.007,
'totalcharges=1901.05': -0.354,
'totalcharges=1901.65': -0.337,
'totalcharges=1908.35': -0.097,
'totalcharges=191.05': 0.536,
'totalcharges=1912.85': -0.121,
'totalcharges=1914.9': 0.349,
'totalcharges=1916': -0.055,
'totalcharges=1923.5': -0.103,
'totalcharges=1924.1': -0.033,
'totalcharges=1927.3': -0.061,
'totalcharges=1929.95': -0.084,
'totalcharges=193.05': -0.131,
'totalcharges=193.6': 0.578,
'totalcharges=1931.3': -0.104,
'totalcharges=1934.45': 0.389,
'totalcharges=1938.05': -0.33,
'totalcharges=1938.9': -0.556,
'totalcharges=1939.35': -0.039,
'totalcharges=194.55': -0.425,
'totalcharges=1940.8': -0.124,
'totalcharges=1940.85': -0.006,
'totalcharges=1941.5': -0.115,
'totalcharges=1943.2': 0.335,
'totalcharges=1949.4': -0.418,
'totalcharges=195.3': 0.427,
'totalcharges=1951': 0.284,
'totalcharges=1952.25': -0.235,
'totalcharges=1952.8': -0.074,
'totalcharges=1955.4': -0.055,
'totalcharges=1957.1': -0.222,
'totalcharges=1958.45': -0.075,
'totalcharges=1958.95': -0.042,
'totalcharges=196.4': -0.542,
'totalcharges=196.75': -0.021,
'totalcharges=196.9': -0.13,
'totalcharges=196.95': 0.474,
'totalcharges=197.7': 0.182,
'totalcharges=1971.15': -0.039,
'totalcharges=198.1': -0.099,
'totalcharges=198.6': -0.13,
'totalcharges=198.7': -0.101,
'totalcharges=1982.1': -0.015,
'totalcharges=199.75': -0.38,
'totalcharges=1990.5': -0.121,
'totalcharges=1992.2': -0.238,
'totalcharges=1992.95': -0.194,
'totalcharges=1993.25': -0.441,
'totalcharges=1993.8': 0.316,
'totalcharges=1994.3': 0.808,
'totalcharges=20.05': 0.508,
'totalcharges=20.1': 1.507,
'totalcharges=20.15': 1.291,
'totalcharges=20.2': 1.034,
'totalcharges=20.25': 0.591,
'totalcharges=20.3': 0.181,

'totalcharges=20.35': -0.366,
'totalcharges=20.4': 0.725,
'totalcharges=20.45': -0.377,
'totalcharges=20.5': 1.549,
'totalcharges=20.55': 0.746,
'totalcharges=20.6': 0.546,
'totalcharges=20.7': -0.176,
'totalcharges=20.8': -0.13,
'totalcharges=20.9': 0.528,
'totalcharges=20.95': -0.14,
'totalcharges=2001': -0.409,
'totalcharges=2006.1': -0.532,
'totalcharges=2006.95': -0.008,
'totalcharges=2007.25': -0.395,
'totalcharges=2007.85': -0.314,
'totalcharges=201': 0.272,
'totalcharges=201.1': 0.51,
'totalcharges=201.95': -0.24,
'totalcharges=2010.55': 0.688,
'totalcharges=2011.4': 0.465,
'totalcharges=2012.7': 0.359,
'totalcharges=2015.35': 0.547,
'totalcharges=2015.8': -0.318,
'totalcharges=202.15': -0.324,
'totalcharges=202.3': -0.267,
'totalcharges=202.9': -0.477,
'totalcharges=2021.2': -0.04,
'totalcharges=2023.55': -0.085,
'totalcharges=2024.1': -0.41,
'totalcharges=2025.1': -0.624,
'totalcharges=2028.8': -0.117,
'totalcharges=2029.05': -0.058,
'totalcharges=2030.75': -0.126,
'totalcharges=2031.95': -0.485,
'totalcharges=2032.3': -0.25,
'totalcharges=2033.05': -0.201,
'totalcharges=2034.25': -0.234,
'totalcharges=2038.7': -0.149,
'totalcharges=204.55': -0.163,
'totalcharges=204.7': 0.206,
'totalcharges=2042.05': -0.053,
'totalcharges=2043.45': 0.298,
'totalcharges=2044.75': -0.071,
'totalcharges=2044.95': -0.384,
'totalcharges=2045.55': -0.133,
'totalcharges=2049.05': -0.105,
'totalcharges=2053.05': -0.086,
'totalcharges=2058.5': -0.121,
'totalcharges=206.15': -0.395,
'totalcharges=206.6': 0.507,
'totalcharges=2065.15': -0.12,
'totalcharges=2065.4': -0.236,
'totalcharges=2066': -0.074,
'totalcharges=2068.55': 0.567,
'totalcharges=2070.75': -0.121,
'totalcharges=2072.75': -0.447,
'totalcharges=2076.2': 0.539,
'totalcharges=208': -0.099,
'totalcharges=208.7': 0.456,
'totalcharges=2080.1': -0.23,
'totalcharges=2085.45': -0.38,
'totalcharges=2088.05': -0.467,
'totalcharges=2088.8': 0.415,
'totalcharges=209.1': 0.331,
'totalcharges=2090.25': -0.297,
'totalcharges=2093.4': -0.373,

```
 'totalcharges=2093.9': -0.087,
 'totalcharges=2095': 0.302,
 'totalcharges=2096.1': -0.239,
 'totalcharges=21.1': -0.13,
 'totalcharges=210.3': 0.214,
 'totalcharges=210.65': -0.031,
 'totalcharges=210.75': 0.364,
 'totalcharges=2104.55': 0.283,
 'totalcharges=2106.05': -0.115,
 'totalcharges=2106.3': -0.104,
 'totalcharges=2108.35': -0.12,
 'totalcharges=2109.35': 0.515,
 'totalcharges=2110.15': 0.373,
 'totalcharges=2117.2': -0.051,
 'totalcharges=2119.5': 0.214,
 'totalcharges=212.3': -0.029,
 'totalcharges=212.4': -0.615,
 'totalcharges=2122.05': 0.343,
 'totalcharges=2122.45': 0.328,
 'totalcharges=213.35': -0.065,
 'totalcharges=2134.3': -0.06,
 'totalcharges=2139.1': -0.159,
 'totalcharges=214.4': -0.347,
 'totalcharges=214.55': -0.375,
 'totalcharges=214.75': -0.36,
 'totalcharges=2142.8': -0.328,
 'totalcharges=2145': 0.719,
 'totalcharges=2146.5': -0.433,
 'totalcharges=2149.05': -0.435,
 'totalcharges=215.25': -0.099,
 'totalcharges=215.8': 0.302,
 'totalcharges=2156.25': -0.11,
 'totalcharges=216.2': -0.096,
 'totalcharges=2162.6': -0.596,
 'totalcharges=2168.15': -0.232,
 'totalcharges=2168.9': -0.107,
 'totalcharges=2169.4': -0.084,
 'totalcharges=2169.8': 0.393,
 'totalcharges=217.1': -0.238,
 'totalcharges=217.45': -0.54,
 'totalcharges=217.55': 0.443,
 'totalcharges=2172.05': -0.502,
 'totalcharges=2180.55': -0.366,
 'totalcharges=2181.55': -0.1,
 'totalcharges=2184.6': -0.203,
 'totalcharges=2184.85': -0.582,
 'totalcharges=2187.15': -0.418,
 'totalcharges=2188.45': -0.069,
 ...}
```

Tp understand how the model works and to build the intution. let's train a simpler and smaller model that uses only three variables: congtract, tenure, and totalcharges.

we will redo the same steps we did for training, this time using a smaller set of features. (no additional preprocessing for tenure and totalcharges since they are numneric. contract is categorical so we need to apply one-hot encoding)

In [45]:

```
subset = ['contract', 'tenure', 'totalcharges']
train_dict_small = df_train[subset].to_dict(orient='records')
dv_small = DictVectorizer(sparse=False)
dv_small.fit(train_dict_small)

X_small_train = dv_small.transform(train_dict_small)
```

```
dv_small.get_feature_names()
```

```
/Users/ryantalbot/opt/anaconda3/envs/tf2/lib/python3.9/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[45]:

```
['contract=month-to-month',
 'contract=one_year',
 'contract=two_year',
 'tenure',
 'totalcharges=100.25',
 'totalcharges=100.4',
 'totalcharges=100.9',
 'totalcharges=1001.5',
 'totalcharges=1004.5',
 'totalcharges=1004.75',
 'totalcharges=1006.9',
 'totalcharges=1007.8',
 'totalcharges=1007.9',
 'totalcharges=1008.55',
 'totalcharges=1008.7',
 'totalcharges=1009.25',
 'totalcharges=101.1',
 'totalcharges=101.9',
 'totalcharges=1011.05',
 'totalcharges=1011.5',
 'totalcharges=1011.8',
 'totalcharges=1012.4',
 'totalcharges=1013.2',
 'totalcharges=1013.6',
 'totalcharges=1016.7',
 'totalcharges=1017.35',
 'totalcharges=102.45',
 'totalcharges=102.75',
 'totalcharges=1020.2',
 'totalcharges=1021.55',
 'totalcharges=1021.75',
 'totalcharges=1022.5',
 'totalcharges=1022.6',
 'totalcharges=1022.95',
 'totalcharges=1023.75',
 'totalcharges=1023.9',
 'totalcharges=1024',
 'totalcharges=1024.65',
 'totalcharges=1025.15',
 'totalcharges=1025.95',
 'totalcharges=1026.35',
 'totalcharges=1028.75',
 'totalcharges=1029.35',
 'totalcharges=1029.8',
 'totalcharges=103.7',
 'totalcharges=1032',
 'totalcharges=1032.05',
 'totalcharges=1036',
 'totalcharges=1036.75',
 'totalcharges=1041.8',
 'totalcharges=1042.65',
 'totalcharges=1043.3',
 'totalcharges=1043.35',
 'totalcharges=1043.8',
 'totalcharges=1045.25',
 'totalcharges=1046.1',
 'totalcharges=1046.2',
```

```
'totalcharges=1046.5',
'totalcharges=1047.7',
'totalcharges=1049.05',
'totalcharges=1051.9',
'totalcharges=1052.35',
'totalcharges=1054.75',
'totalcharges=1055.9',
'totalcharges=1057.85',
'totalcharges=1058.1',
'totalcharges=1058.6',
'totalcharges=1059.55',
'totalcharges=106.2',
'totalcharges=106.8',
'totalcharges=106.9',
'totalcharges=1060.2',
'totalcharges=1061.6',
'totalcharges=1064.65',
'totalcharges=1066.15',
'totalcharges=1066.9',
'totalcharges=1067.15',
'totalcharges=1067.65',
'totalcharges=107.05',
'totalcharges=107.1',
'totalcharges=107.6',
'totalcharges=1070.15',
'totalcharges=1070.25',
'totalcharges=1070.5',
'totalcharges=1070.7',
'totalcharges=1071.4',
'totalcharges=1072',
'totalcharges=1072.6',
'totalcharges=1073.3',
'totalcharges=1077.5',
'totalcharges=1078.9',
'totalcharges=1079.45',
'totalcharges=1079.65',
'totalcharges=108.15',
'totalcharges=1083.7',
'totalcharges=1086.75',
'totalcharges=1087.25',
'totalcharges=1087.7',
'totalcharges=1088.25',
'totalcharges=109.2',
'totalcharges=109.25',
'totalcharges=109.5',
'totalcharges=109.8',
'totalcharges=1090.65',
'totalcharges=1093',
'totalcharges=1093.1',
'totalcharges=1093.2',
'totalcharges=1095.3',
'totalcharges=1095.65',
'totalcharges=1096.25',
'totalcharges=1096.65',
'totalcharges=1098.85',
'totalcharges=110.15',
'totalcharges=1101.85',
'totalcharges=1103.25',
'totalcharges=1107.2',
'totalcharges=1107.25',
'totalcharges=1108',
'totalcharges=1108.6',
'totalcharges=111.05',
'totalcharges=1110.05',
'totalcharges=1110.35',
'totalcharges=1110.5',
```

```
'totalcharges=1111.2',
'totalcharges=1112.3',
'totalcharges=1114.55',
'totalcharges=1114.85',
'totalcharges=1115.15',
'totalcharges=1119.35',
'totalcharges=112.3',
'totalcharges=112.75',
'totalcharges=1120.3',
'totalcharges=1120.95',
'totalcharges=1121.05',
'totalcharges=1122.4',
'totalcharges=1123.15',
'totalcharges=1123.65',
'totalcharges=1125.6',
'totalcharges=1126.75',
'totalcharges=1129.35',
'totalcharges=1129.75',
'totalcharges=113.1',
'totalcharges=113.35',
'totalcharges=113.85',
'totalcharges=1130.85',
'totalcharges=1131.2',
'totalcharges=1132.35',
'totalcharges=1132.75',
'totalcharges=1133.7',
'totalcharges=1134.25',
'totalcharges=1135.7',
'totalcharges=1137.05',
'totalcharges=1138.8',
'totalcharges=114.1',
'totalcharges=1144.5',
'totalcharges=1144.6',
'totalcharges=1145.35',
'totalcharges=1145.7',
'totalcharges=1146.05',
'totalcharges=1146.65',
'totalcharges=1149.65',
'totalcharges=115.1',
'totalcharges=1151.05',
'totalcharges=1151.55',
'totalcharges=1155.6',
'totalcharges=1156.35',
'totalcharges=1157.05',
'totalcharges=1158.85',
'totalcharges=116.85',
'totalcharges=1160.45',
'totalcharges=1161.75',
'totalcharges=1164.05',
'totalcharges=1164.3',
'totalcharges=1165.6',
'totalcharges=1165.9',
'totalcharges=1166.7',
'totalcharges=1169.35',
'totalcharges=117.8',
'totalcharges=1171.3',
'totalcharges=1171.5',
'totalcharges=1173.35',
'totalcharges=1174.35',
'totalcharges=1174.8',
'totalcharges=1175.85',
'totalcharges=1177.05',
'totalcharges=1177.95',
'totalcharges=1178.25',
'totalcharges=1178.4',
'totalcharges=118.3',
```

```
'totalcharges=118.5',
'totalcharges=1180.95',
'totalcharges=1182.55',
'totalcharges=1183.05',
'totalcharges=1183.8',
'totalcharges=1184',
'totalcharges=1188.25',
'totalcharges=1189.9',
'totalcharges=119.75',
'totalcharges=1191.4',
'totalcharges=1192.3',
'totalcharges=1192.7',
'totalcharges=1193.05',
'totalcharges=1195.25',
'totalcharges=1195.95',
'totalcharges=1199.4',
'totalcharges=120.25',
'totalcharges=1200.15',
'totalcharges=1201.65',
'totalcharges=1203.9',
'totalcharges=1203.95',
'totalcharges=1205.05',
'totalcharges=1207',
'totalcharges=1208.15',
'totalcharges=1208.6',
'totalcharges=1209.25',
'totalcharges=121.25',
'totalcharges=1210.3',
'totalcharges=1210.4',
'totalcharges=1215.45',
'totalcharges=1215.65',
'totalcharges=1216.35',
'totalcharges=1218.25',
'totalcharges=1218.55',
'totalcharges=1218.65',
'totalcharges=122',
'totalcharges=122.9',
'totalcharges=1221.65',
'totalcharges=1222.05',
'totalcharges=1222.25',
'totalcharges=1222.65',
'totalcharges=1225.65',
'totalcharges=123.65',
'totalcharges=1230.25',
'totalcharges=1233.4',
'totalcharges=1234.8',
'totalcharges=1235.55',
'totalcharges=1236.15',
'totalcharges=1237.3',
'totalcharges=124.4',
'totalcharges=124.45',
'totalcharges=1240.15',
'totalcharges=1242.2',
'totalcharges=1244.8',
'totalcharges=1245.05',
'totalcharges=1248.9',
'totalcharges=1249.25',
'totalcharges=125.95',
'totalcharges=1252.85',
'totalcharges=1253.5',
'totalcharges=1253.9',
'totalcharges=1258.3',
'totalcharges=1259',
'totalcharges=1259.35',
'totalcharges=126.05',
'totalcharges=1261.7',
```

```
'totalcharges=1263.9',
'totalcharges=1264.2',
'totalcharges=1265.65',
'totalcharges=1266.1',
'totalcharges=1266.35',
'totalcharges=1266.4',
'totalcharges=1267',
'totalcharges=1267.95',
'totalcharges=1268.85',
'totalcharges=127.1',
'totalcharges=1270.55',
'totalcharges=1271.8',
'totalcharges=1274.45',
'totalcharges=1275.6',
'totalcharges=1275.7',
'totalcharges=1277.75',
'totalcharges=128.6',
'totalcharges=1281',
'totalcharges=1282.85',
'totalcharges=1284.2',
'totalcharges=1285.05',
'totalcharges=1288.3',
'totalcharges=1288.75',
'totalcharges=129.55',
'totalcharges=129.6',
'totalcharges=1291.35',
'totalcharges=1292.6',
'totalcharges=1295.4',
'totalcharges=1296.15',
'totalcharges=1297.35',
'totalcharges=1297.8',
'totalcharges=1298.7',
'totalcharges=130.5',
'totalcharges=130.55',
'totalcharges=1301',
'totalcharges=1301.1',
'totalcharges=1301.7',
'totalcharges=1303.5',
'totalcharges=1304.8',
'totalcharges=1304.85',
'totalcharges=1305.95',
'totalcharges=1306.3',
'totalcharges=1308.1',
'totalcharges=1309',
'totalcharges=131.05',
'totalcharges=131.65',
'totalcharges=1311.6',
'totalcharges=1311.75',
'totalcharges=1312.45',
'totalcharges=1313.55',
'totalcharges=1316.9',
'totalcharges=1319.85',
'totalcharges=132.4',
'totalcharges=1322.55',
'totalcharges=1323.7',
'totalcharges=1326.35',
'totalcharges=1327.85',
'totalcharges=1329.2',
'totalcharges=1331.05',
'totalcharges=1332.4',
'totalcharges=1336.1',
'totalcharges=1336.15',
'totalcharges=1336.35',
'totalcharges=1336.9',
'totalcharges=1337.45',
'totalcharges=1337.5',
```

```
'totalcharges=1339.8',
'totalcharges=134.05',
'totalcharges=134.35',
'totalcharges=134.5',
'totalcharges=1340.1',
'totalcharges=1344.5',
'totalcharges=1345.55',
'totalcharges=1345.85',
'totalcharges=1346.9',
'totalcharges=1348.95',
'totalcharges=135.75',
'totalcharges=1355.45',
'totalcharges=1356.3',
'totalcharges=1357.1',
'totalcharges=1362.85',
'totalcharges=1364.3',
'totalcharges=1367.75',
'totalcharges=137.1',
'totalcharges=137.25',
'totalcharges=137.6',
'totalcharges=137.85',
'totalcharges=1370.35',
'totalcharges=1372.45',
'totalcharges=1372.9',
'totalcharges=1373.05',
'totalcharges=1374.2',
'totalcharges=1374.9',
'totalcharges=1375.4',
'totalcharges=1375.6',
'totalcharges=1378.75',
'totalcharges=1379.6',
'totalcharges=138.85',
'totalcharges=1380.1',
'totalcharges=1380.4',
'totalcharges=1381.2',
'totalcharges=1381.8',
'totalcharges=1385.85',
'totalcharges=1386.8',
'totalcharges=1387',
'totalcharges=1387.45',
'totalcharges=1388.45',
'totalcharges=139.05',
'totalcharges=139.25',
'totalcharges=139.4',
'totalcharges=139.75',
'totalcharges=1391.15',
'totalcharges=1391.65',
'totalcharges=1393.6',
'totalcharges=1396',
'totalcharges=1396.9',
'totalcharges=1397.65',
'totalcharges=1398.25',
'totalcharges=140.4',
'totalcharges=140.7',
'totalcharges=1400.55',
'totalcharges=1400.85',
'totalcharges=1401.15',
'totalcharges=1402.25',
'totalcharges=1403.1',
'totalcharges=1404.65',
'totalcharges=1406.9',
'totalcharges=141.1',
'totalcharges=141.45',
'totalcharges=1410.25',
'totalcharges=1411.2',
'totalcharges=1411.35',
```

```
'totalcharges=1411.9',
'totalcharges=1412.4',
'totalcharges=1414.2',
'totalcharges=1414.65',
'totalcharges=1415.55',
'totalcharges=1416.2',
'totalcharges=1416.5',
'totalcharges=1416.75',
'totalcharges=1419.4',
'totalcharges=142.35',
'totalcharges=1421.9',
'totalcharges=1422.1',
'totalcharges=1423.15',
'totalcharges=1423.35',
'totalcharges=1424.2',
'totalcharges=1424.4',
'totalcharges=1424.5',
'totalcharges=1424.6',
'totalcharges=1424.9',
'totalcharges=1424.95',
'totalcharges=1425.45',
'totalcharges=1426.4',
'totalcharges=1427.55',
'totalcharges=143.35',
'totalcharges=1431.65',
'totalcharges=1432.55',
'totalcharges=1434.1',
'totalcharges=1438.05',
'totalcharges=144',
'totalcharges=144.15',
'totalcharges=144.35',
'totalcharges=144.55',
'totalcharges=144.95',
'totalcharges=1441.65',
'totalcharges=1441.95',
'totalcharges=1442.2',
'totalcharges=1442.6',
'totalcharges=1443.65',
'totalcharges=1444.05',
'totalcharges=1444.65',
'totalcharges=1445.2',
'totalcharges=1447.9',
'totalcharges=1448.8',
'totalcharges=145.15',
'totalcharges=1451.1',
'totalcharges=1451.9',
'totalcharges=1453.1',
'totalcharges=1454.25',
'totalcharges=1457.25',
'totalcharges=1458.1',
'totalcharges=1459.35',
'totalcharges=146.05',
'totalcharges=146.4',
'totalcharges=146.6',
'totalcharges=146.9',
'totalcharges=1460.85',
'totalcharges=1461.45',
'totalcharges=1462.05',
'totalcharges=1462.6',
'totalcharges=1463.45',
'totalcharges=1465.75',
'totalcharges=1468.75',
'totalcharges=147.15',
'totalcharges=147.55',
'totalcharges=147.75',
'totalcharges=147.8',
```

```
'totalcharges=1470.05',
'totalcharges=1470.95',
'totalcharges=1471.75',
'totalcharges=1477.65',
'totalcharges=1478.85',
'totalcharges=1483.25',
'totalcharges=1489.3',
'totalcharges=149.05',
'totalcharges=149.55',
'totalcharges=1490.4',
'totalcharges=1490.95',
'totalcharges=1492.1',
'totalcharges=1493.1',
'totalcharges=1493.2',
'totalcharges=1493.55',
'totalcharges=1497.05',
'totalcharges=1497.9',
'totalcharges=1498.2',
'totalcharges=1498.55',
'totalcharges=1498.65',
'totalcharges=1498.85',
'totalcharges=150.35',
'totalcharges=150.6',
'totalcharges=150.75',
'totalcharges=150.85',
'totalcharges=1500.5',
'totalcharges=1500.95',
'totalcharges=1501.75',
'totalcharges=1502.25',
'totalcharges=1504.05',
'totalcharges=1505.05',
'totalcharges=1505.9',
'totalcharges=1506.4',
'totalcharges=1507',
'totalcharges=151.3',
'totalcharges=151.75',
'totalcharges=1510.3',
'totalcharges=1513.6',
'totalcharges=1514.85',
'totalcharges=1515.1',
'totalcharges=1516.6',
'totalcharges=1517.5',
'totalcharges=152.3',
'totalcharges=152.45',
'totalcharges=152.6',
'totalcharges=152.95',
'totalcharges=1520.1',
'totalcharges=1520.9',
'totalcharges=1521.2',
'totalcharges=1522.7',
'totalcharges=1524.85',
'totalcharges=1525.35',
'totalcharges=1527.35',
'totalcharges=1529.2',
'totalcharges=1529.65',
'totalcharges=153.3',
'totalcharges=1531.4',
'totalcharges=1532.45',
'totalcharges=1533.8',
'totalcharges=1534.05',
'totalcharges=1534.75',
'totalcharges=1536.75',
'totalcharges=1537.85',
'totalcharges=1537.9',
'totalcharges=1539.8',
'totalcharges=154.55',
```

```
'totalcharges=154.65',
'totalcharges=154.85',
'totalcharges=1540.05',
'totalcharges=1540.2',
'totalcharges=1540.35',
'totalcharges=1546.3',
'totalcharges=155.35',
'totalcharges=155.8',
'totalcharges=155.9',
'totalcharges=1551.6',
'totalcharges=1553.2',
'totalcharges=1554.9',
'totalcharges=1556.85',
'totalcharges=1559.25',
'totalcharges=1559.45',
'totalcharges=156.25',
'totalcharges=1563.95',
'totalcharges=1566.75',
'totalcharges=1567',
'totalcharges=157.65',
'totalcharges=1573.95',
'totalcharges=1574.45',
'totalcharges=1574.5',
'totalcharges=158.35',
'totalcharges=1584.8',
'totalcharges=1587.55',
'totalcharges=1588.7',
'totalcharges=159.15',
'totalcharges=159.35',
'totalcharges=159.4',
'totalcharges=159.45',
'totalcharges=1595.5',
'totalcharges=1596.6',
'totalcharges=160.05',
'totalcharges=160.75',
'totalcharges=160.85',
'totalcharges=1600.25',
'totalcharges=1600.95',
'totalcharges=1601.5',
'totalcharges=1604.5',
'totalcharges=1607.2',
'totalcharges=161.45',
'totalcharges=161.5',
'totalcharges=161.65',
'totalcharges=1611.65',
'totalcharges=1612.75',
'totalcharges=1614.05',
'totalcharges=1614.7',
'totalcharges=1615.1',
'totalcharges=1618.2',
'totalcharges=162.15',
'totalcharges=162.45',
'totalcharges=1620.2',
'totalcharges=1620.25',
'totalcharges=1621.35',
'totalcharges=1623.4',
'totalcharges=1629.2',
'totalcharges=163.7',
'totalcharges=1630.4',
'totalcharges=1636.95',
'totalcharges=1637.3',
'totalcharges=1638.7',
'totalcharges=164.5',
'totalcharges=164.6',
'totalcharges=164.85',
'totalcharges=1640',
```

```
'totalcharges=1641.3',
'totalcharges=1641.8',
'totalcharges=1642.75',
'totalcharges=165',
'totalcharges=165.35',
'totalcharges=165.45',
'totalcharges=165.5',
'totalcharges=1652.4',
'totalcharges=1652.95',
'totalcharges=1653.45',
'totalcharges=1654.75',
'totalcharges=166.3',
'totalcharges=1660',
'totalcharges=1662.05',
'totalcharges=1663.5',
'totalcharges=1663.75',
'totalcharges=1664.3',
'totalcharges=1665.2',
'totalcharges=1667.25',
'totalcharges=1669.4',
'totalcharges=167.2',
'totalcharges=167.5',
'totalcharges=1671.6',
'totalcharges=1672.1',
'totalcharges=1672.15',
'totalcharges=1672.35',
'totalcharges=1673.4',
'totalcharges=1673.8',
'totalcharges=1677.85',
'totalcharges=1678.05',
'totalcharges=1679.65',
'totalcharges=168.15',
'totalcharges=168.2',
'totalcharges=168.5',
'totalcharges=168.65',
'totalcharges=1680.25',
'totalcharges=1681.6',
'totalcharges=1682.05',
'totalcharges=1682.4',
'totalcharges=1686.15',
'totalcharges=1686.85',
'totalcharges=169.45',
'totalcharges=1692.6',
'totalcharges=1696.2',
'totalcharges=1697.7',
'totalcharges=170.5',
'totalcharges=170.9',
'totalcharges=1700.9',
'totalcharges=1704.95',
'totalcharges=1706.45',
'totalcharges=171',
'totalcharges=171.15',
'totalcharges=1710.15',
'totalcharges=1712.7',
'totalcharges=1714.95',
'totalcharges=1716.45',
'totalcharges=1718.95',
'totalcharges=1719.15',
'totalcharges=172.35',
'totalcharges=172.85',
'totalcharges=1724.15',
'totalcharges=1725.95',
'totalcharges=1727.5',
'totalcharges=1729.35',
'totalcharges=173',
'totalcharges=173.15',
```

```
'totalcharges=1730.65',
'totalcharges=1732.6',
'totalcharges=1734.5',
'totalcharges=1734.65',
'totalcharges=1738.9',
'totalcharges=174.2',
'totalcharges=174.3',
'totalcharges=174.65',
'totalcharges=174.7',
'totalcharges=174.8',
'totalcharges=1742.45',
'totalcharges=1742.5',
'totalcharges=1742.75',
'totalcharges=1747.2',
'totalcharges=1748.55',
'totalcharges=1748.9',
'totalcharges=1750.85',
'totalcharges=1752.45',
'totalcharges=1752.55',
'totalcharges=1752.65',
'totalcharges=1753',
'totalcharges=1755.35',
'totalcharges=1756.6',
'totalcharges=1759.4',
'totalcharges=176.2',
'totalcharges=1766.75',
'totalcharges=1767.35',
'totalcharges=177.4',
'totalcharges=1775.8',
'totalcharges=1776',
'totalcharges=1776.95',
'totalcharges=1777.6',
'totalcharges=1777.9',
'totalcharges=1778.7',
'totalcharges=1779.95',
'totalcharges=178.5',
'totalcharges=178.7',
'totalcharges=1782',
'totalcharges=1783.75',
'totalcharges=1784.5',
'totalcharges=1784.9',
'totalcharges=1789.25',
'totalcharges=179.35',
'totalcharges=1790.15',
'totalcharges=1790.35',
'totalcharges=1790.6',
'totalcharges=1793.25',
'totalcharges=1794.8',
'totalcharges=1796.55',
'totalcharges=1797.1',
'totalcharges=1797.75',
'totalcharges=1798.9',
'totalcharges=18.8',
'totalcharges=18.85',
'totalcharges=18.9',
'totalcharges=1801.1',
'totalcharges=1802.15',
'totalcharges=1803.7',
'totalcharges=1808.7',
'totalcharges=181.1',
'totalcharges=181.5',
'totalcharges=181.6',
'totalcharges=181.65',
'totalcharges=181.7',
'totalcharges=1810.55',
'totalcharges=1816.2',
```

```
'totalcharges=1818.3',
'totalcharges=1821.8',
'totalcharges=1821.95',
'totalcharges=1825.5',
'totalcharges=183.15',
'totalcharges=183.75',
'totalcharges=1830.05',
'totalcharges=1830.1',
'totalcharges=1832.4',
'totalcharges=184.15',
'totalcharges=184.4',
'totalcharges=1840.75',
'totalcharges=1842.7',
'totalcharges=1842.8',
'totalcharges=1845.9',
'totalcharges=1849.95',
'totalcharges=185.2',
'totalcharges=185.55',
'totalcharges=185.6',
'totalcharges=1851.45',
'totalcharges=1852.85',
'totalcharges=1855.65',
'totalcharges=1856.4',
'totalcharges=1857.25',
'totalcharges=1857.3',
'totalcharges=1859.1',
'totalcharges=1859.2',
'totalcharges=1859.5',
'totalcharges=186.15',
'totalcharges=186.3',
'totalcharges=1862.9',
'totalcharges=1863.8',
'totalcharges=1864.65',
'totalcharges=1866.45',
'totalcharges=1867.6',
'totalcharges=1868.4',
'totalcharges=187.45',
'totalcharges=1871.15',
'totalcharges=1871.85',
'totalcharges=1874.3',
'totalcharges=1875.25',
'totalcharges=188.7',
'totalcharges=1880.85',
'totalcharges=1884.65',
'totalcharges=1885.15',
'totalcharges=1886.4',
'totalcharges=1887',
'totalcharges=1889.5',
'totalcharges=189.45',
'totalcharges=189.95',
'totalcharges=1893.95',
'totalcharges=19',
'totalcharges=19.05',
'totalcharges=19.1',
'totalcharges=19.15',
'totalcharges=19.2',
'totalcharges=19.25',
'totalcharges=19.3',
'totalcharges=19.4',
'totalcharges=19.45',
'totalcharges=19.5',
'totalcharges=19.55',
'totalcharges=19.6',
'totalcharges=19.65',
'totalcharges=19.7',
'totalcharges=19.75',
```

```
'totalcharges=19.85',
'totalcharges=19.9',
'totalcharges=19.95',
'totalcharges=190.05',
'totalcharges=190.1',
'totalcharges=190.5',
'totalcharges=1900.25',
'totalcharges=1901',
'totalcharges=1901.05',
'totalcharges=1901.65',
'totalcharges=1908.35',
'totalcharges=191.05',
'totalcharges=1912.85',
'totalcharges=1914.9',
'totalcharges=1916',
'totalcharges=1923.5',
'totalcharges=1924.1',
'totalcharges=1927.3',
'totalcharges=1929.95',
'totalcharges=193.05',
'totalcharges=193.6',
'totalcharges=1931.3',
'totalcharges=1934.45',
'totalcharges=1938.05',
'totalcharges=1938.9',
'totalcharges=1939.35',
'totalcharges=194.55',
'totalcharges=1940.8',
'totalcharges=1940.85',
'totalcharges=1941.5',
'totalcharges=1943.2',
'totalcharges=1949.4',
'totalcharges=195.3',
'totalcharges=1951',
'totalcharges=1952.25',
'totalcharges=1952.8',
'totalcharges=1955.4',
'totalcharges=1957.1',
'totalcharges=1958.45',
'totalcharges=1958.95',
'totalcharges=196.4',
'totalcharges=196.75',
'totalcharges=196.9',
'totalcharges=196.95',
'totalcharges=197.7',
'totalcharges=1971.15',
'totalcharges=198.1',
'totalcharges=198.6',
'totalcharges=198.7',
'totalcharges=1982.1',
'totalcharges=199.75',
'totalcharges=1990.5',
'totalcharges=1992.2',
'totalcharges=1992.95',
'totalcharges=1993.25',
'totalcharges=1993.8',
'totalcharges=1994.3',
'totalcharges=20.05',
'totalcharges=20.1',
'totalcharges=20.15',
'totalcharges=20.2',
'totalcharges=20.25',
'totalcharges=20.3',
'totalcharges=20.35',
'totalcharges=20.4',
'totalcharges=20.45',
```

```
'totalcharges=20.5',
'totalcharges=20.55',
'totalcharges=20.6',
'totalcharges=20.7',
'totalcharges=20.8',
'totalcharges=20.9',
'totalcharges=20.95',
'totalcharges=2001',
'totalcharges=2006.1',
'totalcharges=2006.95',
'totalcharges=2007.25',
'totalcharges=2007.85',
'totalcharges=201',
'totalcharges=201.1',
'totalcharges=201.95',
'totalcharges=2010.55',
'totalcharges=2011.4',
'totalcharges=2012.7',
'totalcharges=2015.35',
'totalcharges=2015.8',
'totalcharges=202.15',
'totalcharges=202.3',
'totalcharges=202.9',
'totalcharges=2021.2',
'totalcharges=2023.55',
'totalcharges=2024.1',
'totalcharges=2025.1',
'totalcharges=2028.8',
'totalcharges=2029.05',
'totalcharges=2030.75',
'totalcharges=2031.95',
'totalcharges=2032.3',
'totalcharges=2033.05',
'totalcharges=2034.25',
'totalcharges=2038.7',
'totalcharges=204.55',
'totalcharges=204.7',
'totalcharges=2042.05',
'totalcharges=2043.45',
'totalcharges=2044.75',
'totalcharges=2044.95',
'totalcharges=2045.55',
'totalcharges=2049.05',
'totalcharges=2053.05',
'totalcharges=2058.5',
'totalcharges=206.15',
'totalcharges=206.6',
'totalcharges=2065.15',
'totalcharges=2065.4',
'totalcharges=2066',
'totalcharges=2068.55',
'totalcharges=2070.75',
'totalcharges=2072.75',
'totalcharges=2076.2',
'totalcharges=208',
'totalcharges=208.7',
'totalcharges=2080.1',
'totalcharges=2085.45',
'totalcharges=2088.05',
'totalcharges=2088.8',
'totalcharges=209.1',
'totalcharges=2090.25',
'totalcharges=2093.4',
'totalcharges=2093.9',
'totalcharges=2095',
'totalcharges=2096.1',
```

```
'totalcharges=21.1',
'totalcharges=210.3',
'totalcharges=210.65',
'totalcharges=210.75',
'totalcharges=2104.55',
'totalcharges=2106.05',
'totalcharges=2106.3',
'totalcharges=2108.35',
'totalcharges=2109.35',
'totalcharges=2110.15',
'totalcharges=2117.2',
'totalcharges=2119.5',
'totalcharges=212.3',
'totalcharges=212.4',
'totalcharges=2122.05',
'totalcharges=2122.45',
'totalcharges=213.35',
'totalcharges=2134.3',
'totalcharges=2139.1',
'totalcharges=214.4',
'totalcharges=214.55',
'totalcharges=214.75',
'totalcharges=2142.8',
'totalcharges=2145',
'totalcharges=2146.5',
'totalcharges=2149.05',
'totalcharges=215.25',
'totalcharges=215.8',
'totalcharges=2156.25',
'totalcharges=216.2',
'totalcharges=2162.6',
'totalcharges=2168.15',
'totalcharges=2168.9',
'totalcharges=2169.4',
'totalcharges=2169.8',
'totalcharges=217.1',
'totalcharges=217.45',
'totalcharges=217.55',
'totalcharges=2172.05',
'totalcharges=2180.55',
'totalcharges=2181.55',
'totalcharges=2184.6',
'totalcharges=2184.85',
'totalcharges=2187.15',
'totalcharges=2188.45',
'totalcharges=2188.5',
'totalcharges=219',
'totalcharges=219.35',
'totalcharges=219.5',
'totalcharges=219.65',
'totalcharges=2193',
'totalcharges=2193.2',
'totalcharges=2193.65',
'totalcharges=2196.15',
'totalcharges=2196.3',
'totalcharges=2196.45',
'totalcharges=2198.3',
'totalcharges=2198.9',
'totalcharges=220.35',
'totalcharges=220.4',
'totalcharges=220.45',
'totalcharges=220.6',
'totalcharges=2200.25',
'totalcharges=2200.7',
'totalcharges=2201.75',
'totalcharges=2203.1',
```

```
 'totalcharges=2203.65',
 'totalcharges=2204.35',
 'totalcharges=2208.05',
 'totalcharges=2208.75',
 'totalcharges=221.1',
 'totalcharges=221.35',
 'totalcharges=2215.4',
 'totalcharges=2217.15',
 'totalcharges=222.3',
 'totalcharges=2224.5',
 'totalcharges=2227.1',
 'totalcharges=2227.8',
 'totalcharges=223.15',
 'totalcharges=223.6',
 'totalcharges=223.75',
 'totalcharges=2230.85',
 'totalcharges=2231.05',
 'totalcharges=2234.55',
 'totalcharges=2234.95',
 ...]
```

In [46]:
```
dv_small.get_feature_names_out()
```

Out[46]:
```
array(['contract=month-to-month', 'contract=one_year',
       'contract=two_year', ..., 'totalcharges=999.45',
       'totalcharges=999.9', 'totalcharges=_'], dtype=object)
```

In [47]:
```
model_small = LogisticRegression(solver='liblinear', random_state=1)
model_small.fit(X_small_train, y_train)
```

Out[47]:
```
LogisticRegression(random_state=1, solver='liblinear')
```

In [48]:
```
model_small.intercept_[0]
```

Out[48]:
```
-0.9485971734124587
```

In [49]:
```
dict(zip(dv_small.get_feature_names(), model_small.coef_[0].round(3)))
```

```
/Users/ryantalbot/opt/anaconda3/envs/tf2/lib/python3.9/site-packages/sklearn/utils/depre
cation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names
is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out instea
d.
  warnings.warn(msg, category=FutureWarning)
```

Out[49]:
```
{'contract=month-to-month': 0.98,
 'contract=one_year': -0.321,
 'contract=two_year': -1.607,
 'tenure': -0.017,
 'totalcharges=100.25': 0.399,
 'totalcharges=100.4': 0.413,
 'totalcharges=100.9': -0.39,
 'totalcharges=1001.5': -0.089,
 'totalcharges=1004.5': -0.098,
 'totalcharges=1004.75': -0.315,
 'totalcharges=1006.9': -0.034,
 'totalcharges=1007.8': -0.094,
 'totalcharges=1007.9': -0.091,
 'totalcharges=1008.55': 0.43,
 'totalcharges=1008.7': 0.408,
 'totalcharges=1009.25': -0.315,
 'totalcharges=101.1': -0.4,
 'totalcharges=101.9': -0.397,
 'totalcharges=1011.05': -0.351,
 'totalcharges=1011.5': -0.341,
 'totalcharges=1011.8': -0.367,
 'totalcharges=1012.4': -0.093,
```

'totalcharges=1013.2': -0.029,
'totalcharges=1013.6': -0.104,
'totalcharges=1016.7': -0.031,
'totalcharges=1017.35': -0.37,
'totalcharges=102.45': 0.399,
'totalcharges=102.75': 0.403,
'totalcharges=1020.2': -0.373,
'totalcharges=1021.55': -0.377,
'totalcharges=1021.75': -0.37,
'totalcharges=1022.5': -0.325,
'totalcharges=1022.6': -0.03,
'totalcharges=1022.95': -0.093,
'totalcharges=1023.75': -0.134,
'totalcharges=1023.9': 0.45,
'totalcharges=1024': -0.05,
'totalcharges=1024.65': -0.03,
'totalcharges=1025.15': -0.315,
'totalcharges=1025.95': 0.443,
'totalcharges=1026.35': -0.089,
'totalcharges=1028.75': -0.029,
'totalcharges=1029.35': -0.367,
'totalcharges=1029.8': -0.031,
'totalcharges=103.7': -0.404,
'totalcharges=1032': 0.488,
'totalcharges=1032.05': -0.097,
'totalcharges=1036': -0.03,
'totalcharges=1036.75': 0.443,
'totalcharges=1041.8': -0.031,
'totalcharges=1042.65': -0.029,
'totalcharges=1043.3': 0.43,
'totalcharges=1043.35': 0.45,
'totalcharges=1043.8': -0.364,
'totalcharges=1045.25': -0.147,
'totalcharges=1046.1': 0.437,
'totalcharges=1046.2': -0.031,
'totalcharges=1046.5': 0.085,
'totalcharges=1047.7': 0.433,
'totalcharges=1049.05': -0.338,
'totalcharges=1051.9': -0.027,
'totalcharges=1052.35': 0.889,
'totalcharges=1054.75': -0.028,
'totalcharges=1055.9': -0.145,
'totalcharges=1057.85': -0.025,
'totalcharges=1058.1': -0.351,
'totalcharges=1058.6': 0.44,
'totalcharges=1059.55': 0.437,
'totalcharges=106.2': -0.404,
'totalcharges=106.8': -0.39,
'totalcharges=106.9': -0.059,
'totalcharges=1060.2': 0.748,
'totalcharges=1061.6': -0.086,
'totalcharges=1064.65': 0.43,
'totalcharges=1066.15': -0.101,
'totalcharges=1066.9': -0.37,
'totalcharges=1067.15': 0.715,
'totalcharges=1067.65': -0.029,
'totalcharges=107.05': -0.393,
'totalcharges=107.1': -0.404,
'totalcharges=107.6': -0.058,
'totalcharges=1070.15': -0.033,
'totalcharges=1070.25': -0.029,
'totalcharges=1070.5': -0.028,
'totalcharges=1070.7': -0.151,
'totalcharges=1071.4': -0.36,
'totalcharges=1072': 0.443,
'totalcharges=1072.6': -0.128,

'totalcharges=1073.3': -0.338,
'totalcharges=1077.5': 0.196,
'totalcharges=1078.9': 0.433,
'totalcharges=1079.45': -0.027,
'totalcharges=1079.65': -0.026,
'totalcharges=108.15': 0.403,
'totalcharges=1083.7': -0.357,
'totalcharges=1086.75': -0.029,
'totalcharges=1087.25': -0.239,
'totalcharges=1087.7': -0.026,
'totalcharges=1088.25': -0.032,
'totalcharges=109.2': -0.39,
'totalcharges=109.25': -0.404,
'totalcharges=109.5': -0.39,
'totalcharges=109.8': -0.393,
'totalcharges=1090.65': -0.093,
'totalcharges=1093': -0.025,
'totalcharges=1093.1': 0.454,
'totalcharges=1093.2': -0.347,
'totalcharges=1095.3': 0.44,
'totalcharges=1095.65': 0.437,
'totalcharges=1096.25': 0.443,
'totalcharges=1096.65': -0.151,
'totalcharges=1098.85': -0.338,
'totalcharges=110.15': -0.39,
'totalcharges=1101.85': 0.443,
'totalcharges=1103.25': -0.025,
'totalcharges=1107.2': -0.028,
'totalcharges=1107.25': 0.447,
'totalcharges=1108': -0.354,
'totalcharges=1108.6': -0.36,
'totalcharges=111.05': -0.404,
'totalcharges=1110.05': -0.028,
'totalcharges=1110.35': -0.028,
'totalcharges=1110.5': -0.104,
'totalcharges=1111.2': -0.354,
'totalcharges=1112.3': -0.364,
'totalcharges=1114.55': -0.36,
'totalcharges=1114.85': -0.05,
'totalcharges=1115.15': 0.685,
'totalcharges=1119.35': -0.139,
'totalcharges=112.3': 0.419,
'totalcharges=112.75': -0.39,
'totalcharges=1120.3': 0.433,
'totalcharges=1120.95': 0.437,
'totalcharges=1121.05': -0.032,
'totalcharges=1122.4': -0.364,
'totalcharges=1123.15': -0.026,
'totalcharges=1123.65': -0.024,
'totalcharges=1125.6': -0.026,
'totalcharges=1126.75': -0.088,
'totalcharges=1129.35': -0.357,
'totalcharges=1129.75': 0.433,
'totalcharges=113.1': -0.404,
'totalcharges=113.35': 0.403,
'totalcharges=113.85': 0.413,
'totalcharges=1130.85': 0.447,
'totalcharges=1131.2': 0.433,
'totalcharges=1132.35': 0.44,
'totalcharges=1132.75': 0.437,
'totalcharges=1133.7': -0.098,
'totalcharges=1134.25': -0.026,
'totalcharges=1135.7': 0.44,
'totalcharges=1137.05': -0.364,
'totalcharges=1138.8': -0.032,
'totalcharges=114.1': 0.403,

'totalcharges=1144.5': 0.454,
'totalcharges=1144.6': -0.347,
'totalcharges=1145.35': 0.433,
'totalcharges=1145.7': 0.447,
'totalcharges=1146.05': -0.028,
'totalcharges=1146.65': 0.447,
'totalcharges=1149.65': -0.027,
'totalcharges=115.1': 0.403,
'totalcharges=1151.05': -0.123,
'totalcharges=1151.55': -0.36,
'totalcharges=1155.6': -0.36,
'totalcharges=1156.35': -0.351,
'totalcharges=1157.05': 0.709,
'totalcharges=1158.85': -0.026,
'totalcharges=116.85': -0.058,
'totalcharges=1160.45': -0.031,
'totalcharges=1161.75': -0.052,
'totalcharges=1164.05': -0.364,
'totalcharges=1164.3': 0.437,
'totalcharges=1165.6': -0.354,
'totalcharges=1165.9': -0.101,
'totalcharges=1166.7': -0.351,
'totalcharges=1169.35': 0.44,
'totalcharges=117.8': -0.178,
'totalcharges=1171.3': -0.032,
'totalcharges=1171.5': -0.098,
'totalcharges=1173.35': -0.03,
'totalcharges=1174.35': 0.464,
'totalcharges=1174.8': -0.101,
'totalcharges=1175.85': -0.297,
'totalcharges=1177.05': -0.024,
'totalcharges=1177.95': -0.027,
'totalcharges=1178.25': 0.45,
'totalcharges=1178.4': 0.443,
'totalcharges=118.3': 0.403,
'totalcharges=118.5': 0.409,
'totalcharges=1180.95': 0.443,
'totalcharges=1182.55': 0.603,
'totalcharges=1183.05': -0.393,
'totalcharges=1183.8': -0.325,
'totalcharges=1184': -0.161,
'totalcharges=1188.25': -0.025,
'totalcharges=1189.9': -0.025,
'totalcharges=119.75': 0.403,
'totalcharges=1191.4': -0.026,
'totalcharges=1192.3': -0.025,
'totalcharges=1192.7': -0.025,
'totalcharges=1193.05': 0.712,
'totalcharges=1195.25': -0.029,
'totalcharges=1195.95': -0.074,
'totalcharges=1199.4': -0.351,
'totalcharges=120.25': -0.404,
'totalcharges=1200.15': 0.437,
'totalcharges=1201.65': -0.151,
'totalcharges=1203.9': -0.364,
'totalcharges=1203.95': -0.098,
'totalcharges=1205.05': -0.025,
'totalcharges=1207': -0.354,
'totalcharges=1208.15': -0.044,
'totalcharges=1208.6': -0.023,
'totalcharges=1209.25': -0.024,
'totalcharges=121.25': -0.061,
'totalcharges=1210.3': -0.027,
'totalcharges=1210.4': 0.478,
'totalcharges=1215.45': -0.143,
'totalcharges=1215.65': -0.367,

'totalcharges=1216.35': -0.094,
'totalcharges=1218.25': -0.331,
'totalcharges=1218.55': 0.437,
'totalcharges=1218.65': -0.023,
'totalcharges=122': -0.393,
'totalcharges=122.9': -0.39,
'totalcharges=1221.65': -0.351,
'totalcharges=1222.05': 0.433,
'totalcharges=1222.25': -0.029,
'totalcharges=1222.65': -0.025,
'totalcharges=1225.65': -0.023,
'totalcharges=123.65': 0.406,
'totalcharges=1230.25': 0.44,
'totalcharges=1233.4': 0.454,
'totalcharges=1234.8': -0.024,
'totalcharges=1235.55': 0.067,
'totalcharges=1236.15': 0.468,
'totalcharges=1237.3': -0.325,
'totalcharges=124.4': -0.173,
'totalcharges=124.45': -0.058,
'totalcharges=1240.15': -0.023,
'totalcharges=1242.2': -0.36,
'totalcharges=1244.8': -0.08,
'totalcharges=1245.05': -0.03,
'totalcharges=1248.9': -0.357,
'totalcharges=1249.25': -0.334,
'totalcharges=125.95': -0.404,
'totalcharges=1252.85': -0.026,
'totalcharges=1253.5': -0.338,
'totalcharges=1253.9': 0.437,
'totalcharges=1258.3': 0.447,
'totalcharges=1259': 0.443,
'totalcharges=1259.35': -0.344,
'totalcharges=126.05': 0.413,
'totalcharges=1261.7': -0.088,
'totalcharges=1263.9': -0.024,
'totalcharges=1264.2': -0.153,
'totalcharges=1265.65': -0.325,
'totalcharges=1266.1': 0.443,
'totalcharges=1266.35': -0.135,
'totalcharges=1266.4': -0.094,
'totalcharges=1267': -0.084,
'totalcharges=1267.95': -0.357,
'totalcharges=1268.85': -0.022,
'totalcharges=127.1': 0.406,
'totalcharges=1270.55': -0.025,
'totalcharges=1271.8': -0.338,
'totalcharges=1274.45': -0.145,
'totalcharges=1275.6': -0.028,
'totalcharges=1275.7': -0.331,
'totalcharges=1277.75': -0.023,
'totalcharges=128.6': -0.171,
'totalcharges=1281': 0.45,
'totalcharges=1282.85': -0.139,
'totalcharges=1284.2': 0.103,
'totalcharges=1285.05': -0.36,
'totalcharges=1288.3': -0.36,
'totalcharges=1288.75': -0.137,
'totalcharges=129.55': -0.39,
'totalcharges=129.6': -0.397,
'totalcharges=1291.35': -0.322,
'totalcharges=1292.6': -0.023,
'totalcharges=1295.4': -0.134,
'totalcharges=1296.15': -0.024,
'totalcharges=1297.35': -0.024,
'totalcharges=1297.8': -0.347,

'totalcharges=1298.7': -0.024,
'totalcharges=130.5': -0.39,
'totalcharges=130.55': -0.387,
'totalcharges=1301': 0.443,
'totalcharges=1301.1': -0.021,
'totalcharges=1301.7': 0.44,
'totalcharges=1303.5': -0.021,
'totalcharges=1304.8': -0.02,
'totalcharges=1304.85': -0.045,
'totalcharges=1305.95': -0.024,
'totalcharges=1306.3': -0.096,
'totalcharges=1308.1': 0.44,
'totalcharges=1309': -0.121,
'totalcharges=131.05': 0.403,
'totalcharges=131.65': 0.403,
'totalcharges=1311.6': -0.024,
'totalcharges=1311.75': -0.022,
'totalcharges=1312.45': -0.334,
'totalcharges=1313.55': 0.454,
'totalcharges=1316.9': -0.036,
'totalcharges=1319.85': -0.331,
'totalcharges=132.4': -0.404,
'totalcharges=1322.55': -0.354,
'totalcharges=1323.7': -0.315,
'totalcharges=1326.35': -0.021,
'totalcharges=1327.85': -0.139,
'totalcharges=1329.2': 0.454,
'totalcharges=1331.05': -0.028,
'totalcharges=1332.4': -0.315,
'totalcharges=1336.1': -0.028,
'totalcharges=1336.15': -0.322,
'totalcharges=1336.35': -0.081,
'totalcharges=1336.9': -0.354,
'totalcharges=1337.45': -0.364,
'totalcharges=1337.5': -0.022,
'totalcharges=1339.8': -0.341,
'totalcharges=134.05': -0.404,
'totalcharges=134.35': 0.403,
'totalcharges=134.5': 0.406,
'totalcharges=1340.1': -0.357,
'totalcharges=1344.5': -0.023,
'totalcharges=1345.55': 0.447,
'totalcharges=1345.85': -0.028,
'totalcharges=1346.9': -0.137,
'totalcharges=1348.95': -0.354,
'totalcharges=135.75': -0.693,
'totalcharges=1355.45': -0.135,
'totalcharges=1356.3': -0.022,
'totalcharges=1357.1': -0.023,
'totalcharges=1362.85': -0.134,
'totalcharges=1364.3': 0.44,
'totalcharges=1367.75': -0.137,
'totalcharges=137.1': -0.404,
'totalcharges=137.25': -0.4,
'totalcharges=137.6': 0.032,
'totalcharges=137.85': 0.409,
'totalcharges=1370.35': -0.344,
'totalcharges=1372.45': -0.021,
'totalcharges=1372.9': -0.022,
'totalcharges=1373.05': -0.351,
'totalcharges=1374.2': -0.023,
'totalcharges=1374.9': -0.357,
'totalcharges=1375.4': -0.022,
'totalcharges=1375.6': -0.347,
'totalcharges=1378.75': -0.021,
'totalcharges=1379.6': -0.021,

'totalcharges=138.85': -0.39,
'totalcharges=1380.1': 0.553,
'totalcharges=1380.4': -0.351,
'totalcharges=1381.2': -0.331,
'totalcharges=1381.8': -0.429,
'totalcharges=1385.85': -0.027,
'totalcharges=1386.8': -0.13,
'totalcharges=1387': -0.351,
'totalcharges=1387.45': -0.021,
'totalcharges=1388.45': -0.021,
'totalcharges=139.05': 0.403,
'totalcharges=139.25': -0.404,
'totalcharges=139.4': 0.403,
'totalcharges=139.75': -0.383,
'totalcharges=1391.15': -0.357,
'totalcharges=1391.65': -0.109,
'totalcharges=1393.6': -0.341,
'totalcharges=1396': -0.022,
'totalcharges=1396.9': -0.022,
'totalcharges=1397.65': -0.021,
'totalcharges=1398.25': -0.357,
'totalcharges=140.4': -0.4,
'totalcharges=140.7': 0.403,
'totalcharges=1400.55': 0.478,
'totalcharges=1400.85': 0.454,
'totalcharges=1401.15': -0.02,
'totalcharges=1402.25': -0.027,
'totalcharges=1403.1': -0.072,
'totalcharges=1404.65': -0.328,
'totalcharges=1406.9': -0.325,
'totalcharges=141.1': -0.387,
'totalcharges=141.45': 0.403,
'totalcharges=1410.25': -0.151,
'totalcharges=1411.2': -0.021,
'totalcharges=1411.35': -0.322,
'totalcharges=1411.9': -0.022,
'totalcharges=1412.4': -0.109,
'totalcharges=1414.2': -0.045,
'totalcharges=1414.65': -0.02,
'totalcharges=1415.55': 0.443,
'totalcharges=1416.2': -0.145,
'totalcharges=1416.5': -0.021,
'totalcharges=1416.75': -0.024,
'totalcharges=1419.4': -0.02,
'totalcharges=142.35': 0.403,
'totalcharges=1421.9': -0.02,
'totalcharges=1422.1': 0.45,
'totalcharges=1423.15': -0.135,
'totalcharges=1423.35': -0.139,
'totalcharges=1424.2': -0.027,
'totalcharges=1424.4': -0.341,
'totalcharges=1424.5': 0.461,
'totalcharges=1424.6': -0.024,
'totalcharges=1424.9': -0.022,
'totalcharges=1424.95': 0.447,
'totalcharges=1425.45': 0.701,
'totalcharges=1426.4': 0.447,
'totalcharges=1427.55': -0.338,
'totalcharges=143.35': 0.403,
'totalcharges=1431.65': -0.088,
'totalcharges=1432.55': -0.02,
'totalcharges=1434.1': -0.02,
'totalcharges=1438.05': -0.351,
'totalcharges=144': -0.404,
'totalcharges=144.15': 0.403,
'totalcharges=144.35': -0.387,

'totalcharges=144.55': 0.403,
'totalcharges=144.95': -0.387,
'totalcharges=1441.65': -0.022,
'totalcharges=1441.95': -0.312,
'totalcharges=1442.2': -0.357,
'totalcharges=1442.6': -0.3,
'totalcharges=1443.65': -0.351,
'totalcharges=1444.05': -0.027,
'totalcharges=1444.65': 0.447,
'totalcharges=1445.2': -0.024,
'totalcharges=1447.9': -0.022,
'totalcharges=1448.8': 0.523,
'totalcharges=145.15': -0.001,
'totalcharges=1451.1': -0.36,
'totalcharges=1451.9': 0.457,
'totalcharges=1453.1': -0.256,
'totalcharges=1454.25': -0.137,
'totalcharges=1457.25': 0.447,
'totalcharges=1458.1': -0.021,
'totalcharges=1459.35': -0.25,
'totalcharges=146.05': 0.403,
'totalcharges=146.4': -0.397,
'totalcharges=146.6': -0.056,
'totalcharges=146.9': 0.403,
'totalcharges=1460.85': -0.02,
'totalcharges=1461.45': -0.322,
'totalcharges=1462.05': -0.021,
'totalcharges=1462.6': -0.303,
'totalcharges=1463.45': 0.429,
'totalcharges=1465.75': 0.461,
'totalcharges=1468.75': -0.02,
'totalcharges=147.15': 0.697,
'totalcharges=147.55': -0.38,
'totalcharges=147.75': -0.4,
'totalcharges=147.8': -0.404,
'totalcharges=1470.05': -0.354,
'totalcharges=1470.95': 0.464,
'totalcharges=1471.75': 0.461,
'totalcharges=1477.65': -0.132,
'totalcharges=1478.85': -0.291,
'totalcharges=1483.25': 0.45,
'totalcharges=1489.3': -0.101,
'totalcharges=149.05': -0.387,
'totalcharges=149.55': 0.403,
'totalcharges=1490.4': -0.338,
'totalcharges=1490.95': 0.461,
'totalcharges=1492.1': -0.02,
'totalcharges=1493.1': -0.025,
'totalcharges=1493.2': -0.341,
'totalcharges=1493.55': -0.344,
'totalcharges=1497.05': -0.143,
'totalcharges=1497.9': -0.085,
'totalcharges=1498.2': 0.712,
'totalcharges=1498.55': -0.309,
'totalcharges=1498.65': -0.042,
'totalcharges=1498.85': -0.022,
'totalcharges=150.35': -0.38,
'totalcharges=150.6': -0.387,
'totalcharges=150.75': -0.387,
'totalcharges=150.85': -0.171,
'totalcharges=1500.5': -0.351,
'totalcharges=1500.95': -0.021,
'totalcharges=1501.75': -0.024,
'totalcharges=1502.25': -0.02,
'totalcharges=1504.05': 0.447,
'totalcharges=1505.05': -0.025,

'totalcharges=1505.9': -0.083,
'totalcharges=1506.4': -0.083,
'totalcharges=1507': -0.134,
'totalcharges=151.3': 0.406,
'totalcharges=151.75': -0.404,
'totalcharges=1510.3': -0.322,
'totalcharges=1513.6': -0.026,
'totalcharges=1514.85': -0.023,
'totalcharges=1515.1': 0.454,
'totalcharges=1516.6': -0.13,
'totalcharges=1517.5': -0.319,
'totalcharges=152.3': -0.4,
'totalcharges=152.45': 0.403,
'totalcharges=152.6': -0.383,
'totalcharges=152.95': -0.387,
'totalcharges=1520.1': -0.039,
'totalcharges=1520.9': 0.464,
'totalcharges=1521.2': -0.02,
'totalcharges=1522.7': -0.357,
'totalcharges=1524.85': -0.046,
'totalcharges=1525.35': -0.115,
'totalcharges=1527.35': 0.457,
'totalcharges=1529.2': 0.45,
'totalcharges=1529.65': -0.02,
'totalcharges=153.3': -0.058,
'totalcharges=1531.4': 0.443,
'totalcharges=1532.45': 0.461,
'totalcharges=1533.8': -0.139,
'totalcharges=1534.05': -0.025,
'totalcharges=1534.75': 0.45,
'totalcharges=1536.75': -0.312,
'totalcharges=1537.85': -0.101,
'totalcharges=1537.9': -0.351,
'totalcharges=1539.8': -0.36,
'totalcharges=154.55': -0.4,
'totalcharges=154.65': 0.406,
'totalcharges=154.85': 0.403,
'totalcharges=1540.05': -0.357,
'totalcharges=1540.2': -0.135,
'totalcharges=1540.35': -0.344,
'totalcharges=1546.3': -0.13,
'totalcharges=155.35': -0.397,
'totalcharges=155.8': 0.403,
'totalcharges=155.9': -0.387,
'totalcharges=1551.6': -0.024,
'totalcharges=1553.2': -0.134,
'totalcharges=1554.9': -0.024,
'totalcharges=1556.85': -0.354,
'totalcharges=1559.25': -0.36,
'totalcharges=1559.45': -0.331,
'totalcharges=156.25': 0.423,
'totalcharges=1563.95': -0.024,
'totalcharges=1566.75': -0.36,
'totalcharges=1567': -0.02,
'totalcharges=157.65': -0.38,
'totalcharges=1573.95': -0.338,
'totalcharges=1574.45': 0.454,
'totalcharges=1574.5': -0.024,
'totalcharges=158.35': -0.383,
'totalcharges=1584.8': -0.023,
'totalcharges=1587.55': 0.45,
'totalcharges=1588.7': -0.312,
'totalcharges=159.15': 0.406,
'totalcharges=159.35': -0.056,
'totalcharges=159.4': 0.403,
'totalcharges=159.45': 0.403,

'totalcharges=1595.5': -0.078,
'totalcharges=1596.6': -0.103,
'totalcharges=160.05': -0.169,
'totalcharges=160.75': 0.403,
'totalcharges=160.85': -0.4,
'totalcharges=1600.25': -0.338,
'totalcharges=1600.95': -0.077,
'totalcharges=1601.5': -0.351,
'totalcharges=1604.5': 0.774,
'totalcharges=1607.2': 0.454,
'totalcharges=161.45': -0.684,
'totalcharges=161.5': 0.403,
'totalcharges=161.65': -0.056,
'totalcharges=1611.65': -0.322,
'totalcharges=1612.75': 0.457,
'totalcharges=1614.05': -0.331,
'totalcharges=1614.7': -0.354,
'totalcharges=1615.1': -0.338,
'totalcharges=1618.2': -0.024,
'totalcharges=162.15': -0.056,
'totalcharges=162.45': 0.403,
'totalcharges=1620.2': -0.022,
'totalcharges=1620.25': -0.264,
'totalcharges=1621.35': -0.242,
'totalcharges=1623.4': -0.093,
'totalcharges=1629.2': -0.022,
'totalcharges=163.7': -0.167,
'totalcharges=1630.4': -0.112,
'totalcharges=1636.95': -0.128,
'totalcharges=1637.3': -0.347,
'totalcharges=1638.7': -0.024,
'totalcharges=164.5': 0.403,
'totalcharges=164.6': -0.167,
'totalcharges=164.85': 0.419,
'totalcharges=1640': -0.354,
'totalcharges=1641.3': -0.344,
'totalcharges=1641.8': -0.022,
'totalcharges=1642.75': -0.388,
'totalcharges=165': 0.403,
'totalcharges=165.35': -0.057,
'totalcharges=165.45': 0.403,
'totalcharges=165.5': -0.056,
'totalcharges=1652.4': -0.325,
'totalcharges=1652.95': -0.322,
'totalcharges=1653.45': -0.306,
'totalcharges=1654.75': -0.023,
'totalcharges=166.3': 0.403,
'totalcharges=1660': 0.461,
'totalcharges=1662.05': -0.037,
'totalcharges=1663.5': -0.021,
'totalcharges=1663.75': -0.141,
'totalcharges=1664.3': -0.022,
'totalcharges=1665.2': -0.303,
'totalcharges=1667.25': -0.155,
'totalcharges=1669.4': -0.36,
'totalcharges=167.2': -0.393,
'totalcharges=167.5': 0.403,
'totalcharges=1671.6': -0.344,
'totalcharges=1672.1': -0.354,
'totalcharges=1672.15': -0.022,
'totalcharges=1672.35': -0.347,
'totalcharges=1673.4': -0.022,
'totalcharges=1673.8': -0.023,
'totalcharges=1677.85': -0.279,
'totalcharges=1678.05': -0.022,
'totalcharges=1679.65': -0.328,

'totalcharges=168.15': 0.406,
'totalcharges=168.2': 0.403,
'totalcharges=168.5': -0.404,
'totalcharges=168.65': -0.387,
'totalcharges=1680.25': -0.022,
'totalcharges=1681.6': -0.021,
'totalcharges=1682.05': -0.13,
'totalcharges=1682.4': -0.347,
'totalcharges=1686.15': -0.022,
'totalcharges=1686.85': -0.147,
'totalcharges=169.45': 0.406,
'totalcharges=1692.6': 0.471,
'totalcharges=1696.2': -0.128,
'totalcharges=1697.7': -0.023,
'totalcharges=170.5': -0.4,
'totalcharges=170.9': -0.383,
'totalcharges=1700.9': -0.112,
'totalcharges=1704.95': 0.699,
'totalcharges=1706.45': -0.338,
'totalcharges=171': -0.173,
'totalcharges=171.15': 0.403,
'totalcharges=1710.15': -0.021,
'totalcharges=1712.7': 0.478,
'totalcharges=1714.95': -0.023,
'totalcharges=1716.45': 0.756,
'totalcharges=1718.95': -0.315,
'totalcharges=1719.15': -0.021,
'totalcharges=172.35': -0.167,
'totalcharges=172.85': 0.403,
'totalcharges=1724.15': -0.021,
'totalcharges=1725.95': -0.147,
'totalcharges=1727.5': 0.464,
'totalcharges=1729.35': -0.328,
'totalcharges=173': -0.387,
'totalcharges=173.15': -0.171,
'totalcharges=1730.65': -0.021,
'totalcharges=1732.6': -0.331,
'totalcharges=1734.5': -0.128,
'totalcharges=1734.65': -0.02,
'totalcharges=1738.9': -0.02,
'totalcharges=174.2': -0.4,
'totalcharges=174.3': 0.409,
'totalcharges=174.65': -0.056,
'totalcharges=174.7': -0.387,
'totalcharges=174.8': 0.403,
'totalcharges=1742.45': -0.143,
'totalcharges=1742.5': -0.354,
'totalcharges=1742.75': 0.471,
'totalcharges=1747.2': -0.341,
'totalcharges=1748.55': -0.023,
'totalcharges=1748.9': -0.096,
'totalcharges=1750.85': -0.303,
'totalcharges=1752.45': -0.354,
'totalcharges=1752.55': 0.457,
'totalcharges=1752.65': -0.328,
'totalcharges=1753': -0.021,
'totalcharges=1755.35': -0.031,
'totalcharges=1756.6': -0.139,
'totalcharges=1759.4': -0.045,
'totalcharges=176.2': -0.397,
'totalcharges=1766.75': -0.021,
'totalcharges=1767.35': 0.481,
'totalcharges=177.4': -0.4,
'totalcharges=1775.8': 0.461,
'totalcharges=1776': -0.02,
'totalcharges=1776.95': 0.457,

'totalcharges=1777.6': -0.02,
'totalcharges=1777.9': 0.461,
'totalcharges=1778.7': -0.02,
'totalcharges=1779.95': 0.457,
'totalcharges=178.5': 0.426,
'totalcharges=178.7': -0.383,
'totalcharges=1782': -0.122,
'totalcharges=1783.75': -0.334,
'totalcharges=1784.5': -0.02,
'totalcharges=1784.9': -0.022,
'totalcharges=1789.25': 0.461,
'totalcharges=179.35': 0.409,
'totalcharges=1790.15': -0.021,
'totalcharges=1790.35': -0.022,
'totalcharges=1790.6': -0.112,
'totalcharges=1793.25': -0.322,
'totalcharges=1794.8': -0.1,
'totalcharges=1796.55': -0.022,
'totalcharges=1797.1': -0.02,
'totalcharges=1797.75': -0.328,
'totalcharges=1798.9': -0.02,
'totalcharges=18.8': -0.184,
'totalcharges=18.85': 0.399,
'totalcharges=18.9': -0.407,
'totalcharges=1801.1': -0.315,
'totalcharges=1802.15': -0.319,
'totalcharges=1803.7': -0.303,
'totalcharges=1808.7': 0.468,
'totalcharges=181.1': 0.403,
'totalcharges=181.5': 0.403,
'totalcharges=181.6': 0.409,
'totalcharges=181.65': -0.001,
'totalcharges=181.7': 0.403,
'totalcharges=1810.55': -0.021,
'totalcharges=1816.2': -0.021,
'totalcharges=1818.3': -0.145,
'totalcharges=1821.8': -0.022,
'totalcharges=1821.95': -0.338,
'totalcharges=1825.5': -0.309,
'totalcharges=183.15': -0.167,
'totalcharges=183.75': 0.403,
'totalcharges=1830.05': -0.02,
'totalcharges=1830.1': -0.306,
'totalcharges=1832.4': -0.137,
'totalcharges=184.15': -0.404,
'totalcharges=184.4': -0.165,
'totalcharges=1840.75': -0.103,
'totalcharges=1842.7': -0.031,
'totalcharges=1842.8': 0.464,
'totalcharges=1845.9': 0.471,
'totalcharges=1849.95': -0.334,
'totalcharges=185.2': -0.165,
'totalcharges=185.55': 0.403,
'totalcharges=185.6': 0.426,
'totalcharges=1851.45': -0.02,
'totalcharges=1852.85': 0.468,
'totalcharges=1855.65': 0.471,
'totalcharges=1856.4': -0.071,
'totalcharges=1857.25': -0.341,
'totalcharges=1857.3': -0.121,
'totalcharges=1859.1': -0.02,
'totalcharges=1859.2': -0.02,
'totalcharges=1859.5': -0.145,
'totalcharges=186.15': -0.38,
'totalcharges=186.3': -0.383,
'totalcharges=1862.9': -0.341,

'totalcharges=1863.8': 0.72,
'totalcharges=1864.65': -0.306,
'totalcharges=1866.45': -0.02,
'totalcharges=1867.6': -0.319,
'totalcharges=1868.4': -0.334,
'totalcharges=187.45': 0.403,
'totalcharges=1871.15': -0.344,
'totalcharges=1871.85': -0.02,
'totalcharges=1874.3': -0.02,
'totalcharges=1875.25': -0.031,
'totalcharges=188.7': 0.406,
'totalcharges=1880.85': 0.529,
'totalcharges=1884.65': -0.303,
'totalcharges=1885.15': -0.12,
'totalcharges=1886.4': -0.112,
'totalcharges=1887': -0.02,
'totalcharges=1889.5': -0.12,
'totalcharges=189.45': -0.165,
'totalcharges=189.95': -0.387,
'totalcharges=1893.95': 0.488,
'totalcharges=19': -0.407,
'totalcharges=19.05': -0.407,
'totalcharges=19.1': -0.704,
'totalcharges=19.15': -0.407,
'totalcharges=19.2': -0.704,
'totalcharges=19.25': -0.704,
'totalcharges=19.3': -0.009,
'totalcharges=19.4': -0.704,
'totalcharges=19.45': -0.009,
'totalcharges=19.5': -0.704,
'totalcharges=19.55': -0.923,
'totalcharges=19.6': 0.691,
'totalcharges=19.65': 0.291,
'totalcharges=19.7': -0.407,
'totalcharges=19.75': -0.84,
'totalcharges=19.85': 0.399,
'totalcharges=19.9': -0.009,
'totalcharges=19.95': 0.291,
'totalcharges=190.05': 0.403,
'totalcharges=190.1': -0.38,
'totalcharges=190.5': -0.404,
'totalcharges=1900.25': 0.474,
'totalcharges=1901': -0.021,
'totalcharges=1901.05': -0.328,
'totalcharges=1901.65': -0.344,
'totalcharges=1908.35': -0.134,
'totalcharges=191.05': 0.409,
'totalcharges=1912.85': -0.273,
'totalcharges=1914.9': 0.471,
'totalcharges=1916': -0.118,
'totalcharges=1923.5': -0.312,
'totalcharges=1924.1': -0.037,
'totalcharges=1927.3': -0.132,
'totalcharges=1929.95': -0.127,
'totalcharges=193.05': -0.377,
'totalcharges=193.6': 0.703,
'totalcharges=1931.3': -0.106,
'totalcharges=1934.45': 0.461,
'totalcharges=1938.05': -0.331,
'totalcharges=1938.9': -0.338,
'totalcharges=1939.35': -0.117,
'totalcharges=194.55': -0.404,
'totalcharges=1940.8': -0.139,
'totalcharges=1940.85': -0.021,
'totalcharges=1941.5': -0.117,
'totalcharges=1943.2': 0.474,

'totalcharges=1949.4': -0.338,
'totalcharges=195.3': 0.409,
'totalcharges=1951': 0.464,
'totalcharges=1952.25': -0.328,
'totalcharges=1952.8': -0.121,
'totalcharges=1955.4': -0.121,
'totalcharges=1957.1': -0.322,
'totalcharges=1958.45': -0.128,
'totalcharges=1958.95': -0.093,
'totalcharges=196.4': -0.4,
'totalcharges=196.75': 0.023,
'totalcharges=196.9': 0.002,
'totalcharges=196.95': 0.409,
'totalcharges=197.7': 0.403,
'totalcharges=1971.15': -0.035,
'totalcharges=198.1': -0.37,
'totalcharges=198.6': -0.377,
'totalcharges=198.7': -0.377,
'totalcharges=1982.1': -0.028,
'totalcharges=199.75': -0.397,
'totalcharges=1990.5': -0.137,
'totalcharges=1992.2': -0.13,
'totalcharges=1992.95': -0.328,
'totalcharges=1993.25': -0.315,
'totalcharges=1993.8': 0.471,
'totalcharges=1994.3': 0.502,
'totalcharges=20.05': -0.006,
'totalcharges=20.1': 0.909,
'totalcharges=20.15': 0.515,
'totalcharges=20.2': 0.217,
'totalcharges=20.25': -0.235,
'totalcharges=20.3': -0.532,
'totalcharges=20.35': -0.923,
'totalcharges=20.4': 0.399,
'totalcharges=20.45': -0.941,
'totalcharges=20.5': 0.909,
'totalcharges=20.55': 0.399,
'totalcharges=20.6': -0.006,
'totalcharges=20.7': -0.407,
'totalcharges=20.8': -0.407,
'totalcharges=20.9': -0.006,
'totalcharges=20.95': -0.407,
'totalcharges=2001': -0.331,
'totalcharges=2006.1': -0.325,
'totalcharges=2006.95': -0.021,
'totalcharges=2007.25': -0.347,
'totalcharges=2007.85': -0.264,
'totalcharges=201': 0.409,
'totalcharges=201.1': 0.409,
'totalcharges=201.95': -0.387,
'totalcharges=2010.55': 0.516,
'totalcharges=2011.4': 0.529,
'totalcharges=2012.7': 0.461,
'totalcharges=2015.35': 0.481,
'totalcharges=2015.8': -0.325,
'totalcharges=202.15': -0.393,
'totalcharges=202.3': -0.397,
'totalcharges=202.9': -0.4,
'totalcharges=2021.2': -0.118,
'totalcharges=2023.55': -0.11,
'totalcharges=2024.1': -0.334,
'totalcharges=2025.1': -0.341,
'totalcharges=2028.8': -0.328,
'totalcharges=2029.05': -0.109,
'totalcharges=2030.75': -0.291,
'totalcharges=2031.95': -0.344,

'totalcharges=2032.3': -0.357,
'totalcharges=2033.05': -0.137,
'totalcharges=2034.25': -0.134,
'totalcharges=2038.7': -0.127,
'totalcharges=204.55': -0.38,
'totalcharges=204.7': 0.406,
'totalcharges=2042.05': -0.12,
'totalcharges=2043.45': 0.481,
'totalcharges=2044.75': -0.127,
'totalcharges=2044.95': -0.334,
'totalcharges=2045.55': -0.306,
'totalcharges=2049.05': -0.107,
'totalcharges=2053.05': -0.12,
'totalcharges=2058.5': -0.282,
'totalcharges=206.15': -0.4,
'totalcharges=206.6': 0.409,
'totalcharges=2065.15': -0.264,
'totalcharges=2065.4': -0.319,
'totalcharges=2066': -0.285,
'totalcharges=2068.55': 0.492,
'totalcharges=2070.75': -0.256,
'totalcharges=2072.75': -0.331,
'totalcharges=2076.2': 0.543,
'totalcharges=208': -0.373,
'totalcharges=208.7': 0.416,
'totalcharges=2080.1': -0.128,
'totalcharges=2085.45': -0.334,
'totalcharges=2088.05': -0.328,
'totalcharges=2088.8': 0.485,
'totalcharges=209.1': 0.406,
'totalcharges=2090.25': -0.141,
'totalcharges=2093.4': -0.322,
'totalcharges=2093.9': -0.04,
'totalcharges=2095': 0.471,
'totalcharges=2096.1': -0.297,
'totalcharges=21.1': -0.407,
'totalcharges=210.3': 0.406,
'totalcharges=210.65': -0.055,
'totalcharges=210.75': 0.413,
'totalcharges=2104.55': 0.471,
'totalcharges=2106.05': -0.256,
'totalcharges=2106.3': -0.12,
'totalcharges=2108.35': -0.259,
'totalcharges=2109.35': 0.533,
'totalcharges=2110.15': 0.468,
'totalcharges=2117.2': -0.223,
'totalcharges=2119.5': 0.468,
'totalcharges=212.3': -0.053,
'totalcharges=212.4': -0.4,
'totalcharges=2122.05': 0.488,
'totalcharges=2122.45': 0.478,
'totalcharges=213.35': -0.061,
'totalcharges=2134.3': -0.072,
'totalcharges=2139.1': -0.27,
'totalcharges=214.4': -0.4,
'totalcharges=214.55': -0.397,
'totalcharges=214.75': -0.393,
'totalcharges=2142.8': -0.338,
'totalcharges=2145': 0.56,
'totalcharges=2146.5': -0.325,
'totalcharges=2149.05': -0.315,
'totalcharges=215.25': -0.373,
'totalcharges=215.8': 0.406,
'totalcharges=2156.25': -0.309,
'totalcharges=216.2': -0.373,
'totalcharges=2162.6': -0.338,

```
 'totalcharges=2168.15': -0.309,
 'totalcharges=2168.9': -0.264,
 'totalcharges=2169.4': -0.273,
 'totalcharges=2169.8': 0.474,
 'totalcharges=217.1': -0.397,
 'totalcharges=217.45': -0.4,
 'totalcharges=217.55': 0.406,
 'totalcharges=2172.05': -0.344,
 'totalcharges=2180.55': -0.338,
 'totalcharges=2181.55': -0.25,
 'totalcharges=2184.6': -0.134,
 'totalcharges=2184.85': -0.334,
 'totalcharges=2187.15': -0.331,
 'totalcharges=2188.45': -0.096,
 'totalcharges=2188.5': -0.117,
 'totalcharges=219': 0.413,
 'totalcharges=219.35': -0.161,
 'totalcharges=219.5': -0.165,
 'totalcharges=219.65': 0.409,
 'totalcharges=2193': -0.325,
 'totalcharges=2193.2': -0.035,
 'totalcharges=2193.65': -0.325,
 'totalcharges=2196.15': 0.481,
 'totalcharges=2196.3': -0.098,
 'totalcharges=2196.45': -0.325,
 'totalcharges=2198.3': -0.045,
 'totalcharges=2198.9': -0.127,
 'totalcharges=220.35': -0.37,
 'totalcharges=220.4': -0.4,
 'totalcharges=220.45': -0.676,
 'totalcharges=220.6': 0.406,
 'totalcharges=2200.25': 0.468,
 'totalcharges=2200.7': -0.035,
 'totalcharges=2201.75': -0.123,
 'totalcharges=2203.1': -0.115,
 'totalcharges=2203.65': 0.54,
 'totalcharges=2204.35': -0.132,
 'totalcharges=2208.05': -0.096,
 'totalcharges=2208.75': -0.325,
 'totalcharges=221.1': -0.4,
 'totalcharges=221.35': 0.433,
 'totalcharges=2215.4': -0.279,
 'totalcharges=2217.15': 0.502,
 'totalcharges=222.3': 0.406,
 'totalcharges=2224.5': -0.112,
 'totalcharges=2227.1': 0.492,
 'totalcharges=2227.8': -0.309,
 'totalcharges=223.15': 0.413,
 'totalcharges=223.6': -0.37,
 'totalcharges=223.75': -0.397,
 'totalcharges=2230.85': -0.096,
 'totalcharges=2231.05': 0.478,
 'totalcharges=2234.55': -0.117,
 'totalcharges=2234.95': -0.338,
 ...}
```

In [50]:
```
val_dict_small = df_val[subset].to_dict(orient='records')
X_small_val = dv_small.transform(val_dict_small)
```

In [51]:
```
y_pred_small = model_small.predict_proba(X_small_val)[:, 1]
```

In [52]:
```
customer = {
    'customerid': '8879-zkjof',
    'gender': 'female',
    'seniorcitizen': 0,
```

```python
    'partner': 'no',
    'dependents': 'no',
    'tenure': 41,
    'phoneservice': 'yes',
    'multiplelines': 'no',
    'internetservice': 'dsl',
    'onlinesecurity': 'yes',
    'onlinebackup': 'no',
    'deviceprotection': 'yes',
    'techsupport': 'yes',
    'streamingtv': 'yes',
    'streamingmovies': 'yes',
    'contract': 'one_year',
    'paperlessbilling': 'yes',
    'paymentmethod': 'bank_transfer_(automatic)',
    'monthlycharges': 79.85,
    'totalcharges': 3320.75,
}
```

In [53]:
```python
X_test = dv.transform([customer])
model.predict_proba(X_test)[0, 1]
```

Out[53]:
```
0.08568837283157685
```

In [54]:
```python
print(list(X_test[0]))
```

```
[0.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 79.85, 1.0, 0.0, 0.0,
1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.
0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 41.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

In [55]:

```python
customer = {
    'gender': 'female',
    'seniorcitizen': 1,
    'partner': 'no',
    'dependents': 'no',
    'phoneservice': 'yes',
    'multiplelines': 'yes',
    'internetservice': 'fiber_optic',
    'onlinesecurity': 'no',
    'onlinebackup': 'no',
    'deviceprotection': 'no',
    'techsupport': 'no',
    'streamingtv': 'yes',
    'streamingmovies': 'no',
    'contract': 'month-to-month',
    'paperlessbilling': 'yes',
    'paymentmethod': 'electronic_check',
    'tenure': 1,
    'monthlycharges': 85.7,
    'totalcharges': 85.7
}
```

In [56]:

```python
X_test = dv.transform([customer])
model.predict_proba(X_test)[0, 1]
```

Out[56]:

```
0.8518713895728706
```

Chapter 4 Evaluation metrics for classification

In [57]:
```
# get predictons from the model
y_pred = model.predict_proba(X_val)[:, 1]
# Makes "hard" predictions
churn = y_pred >= 0.5
# Computes teh accuracy
(churn == y_val).mean()
```

Out[57]:
```
0.8064516129032258
```

In [58]:
```
from sklearn.metrics import accuracy_score
```

In [59]:
```
# create an array with different thresholds: 0.0, 0.1, 0.2, etc.
thresholds = np.linspace(0, 1, 11)

for t in thresholds:
    churn = y_pred >= t
    acc = accuracy_score(y_val, churn)
    print('%0.2f, %0.3f' % (t, acc))
```

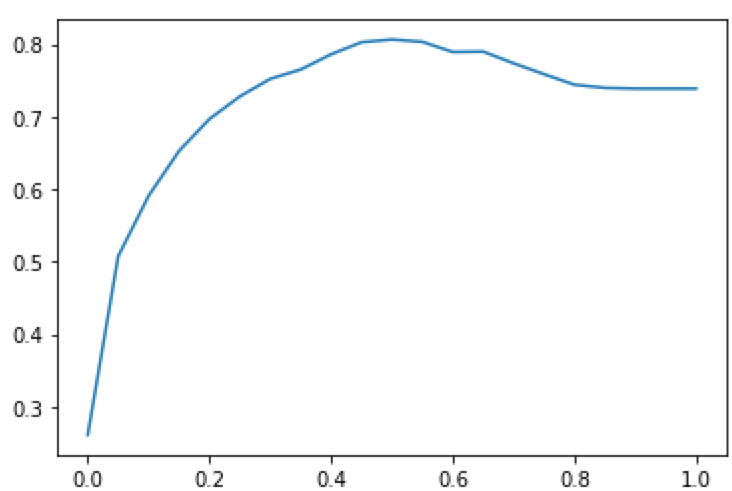
```
0.00, 0.261
0.10, 0.591
0.20, 0.697
0.30, 0.752
0.40, 0.786
0.50, 0.806
0.60, 0.789
0.70, 0.774
0.80, 0.744
0.90, 0.739
1.00, 0.739
```

below example of how accuracy changes depending on the threshold

In [60]:
```
thresholds = np.linspace(0, 1, 21)
accuracies = []
for t in thresholds:
    acc = accuracy_score(y_val, y_pred >= t)
    accuracies.append(acc)
```

In [61]:
```
plt.plot(thresholds, accuracies)
```

Out[61]:
```
[<matplotlib.lines.Line2D at 0x7f9081a4ce80>]
```



To check its accuracy we first make predictions on the validation dataset and then compute the accuracy score

In [62]:
```
val_dict_small = df_val[small_subset].to_dict(orient='records')

X_small_val = dv_small.transform(val_dict_small)
y_pred_small = model_small.predict_proba(X_small_val)[:, 1]

churn_small = y_pred_small >= 0.5
accuracy_score(y_val, churn_small)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [62], in <cell line: 1>()
----> 1 val_dict_small = df_val[small_subset].to_dict(orient='records')
      3 X_small_val = dv_small.transform(val_dict_small)
      4 y_pred_small = model_small.predict_proba(X_small_val)[:, 1]

NameError: name 'small_subset' is not defined
```

Dummy Baseline

baseline prediction

In [63]:
```
size_val = len(y_val)
baseline = np.repeat(False, size_val)
```

In [64]:
```
accuracy_score(baseline, y_val)
```

Out[64]:
```
0.7387096774193549
```

In [65]:
```
t = 0.5
predict_churn = (y_pred >= t)
predict_no_churn = (y_pred < t)

actual_churn = (y_val == 1)
actual_no_churn = (y_val == 0)


true_positive = (predict_churn & actual_churn).sum()
false_positive = (predict_churn & actual_no_churn).sum()

false_negative = (predict_no_churn & actual_churn).sum()
true_negative = (predict_no_churn & actual_no_churn).sum()
```

put above values together in NumPy

In [66]:
```
true_positive = ((y_pred >= 0.5) & (y_val == 1)).sum()
false_positive = ((y_pred >= 0.5) & (y_val == 0)).sum()
false_negative = ((y_pred < 0.5) & (y_val == 1)).sum()
true_negative = ((y_pred < 0.5) & (y_val == 0)).sum()
```

In [67]:
```
confusion_table = np.array(
[[true_negative, false_positive],
[false_negative, true_positive]])
```

In [68]:
```
confusion_table
```

Out[68]:
```
array([[1209,  165],
       [ 195,  291]])
```

In [69]:
```
confusion_table / confusion_table.sum()
```
Out[69]:
```
array([[0.65      , 0.08870968],
       [0.10483871, 0.15645161]])
```
In [70]:
```
precision = true_positive / (true_positive + false_positive)
recall = true_positive / (true_positive + false_negative)
precision, recall
```
Out[70]:
```
(0.6381578947368421, 0.5987654320987654)
```
confusion table not working for small data set

In [71]:
```
# t = 0.5
# s_predict_churn = (y_pred_small >= t)
# s_predict_no_churn = (y_pred_small < t)

# s_actual_churn = (y_val == 1)
# s_actual_no_churn = (y_val == 0)


# s_true_positive = (churn_small & actual_churn).sum()
# s_false_positive = (churn_small & actual_no_churn).sum()

# s_false_negative = (predict_no_churn & actual_churn).sum()
# s_true_negative = (predict_no_churn & actual_no_churn).sum()
```
In [72]:
```
# s_confusion_table = np.array(
# [[s_true_negative, s_false_positive],
# [s_false_negative, s_true_positive]])
```
In [73]:
```
# s_confusion_table
```
accuracy = (TN + TP) / (TN + TP + FN +FP)

Precision and Recall

accuracy can be misleading when dealing with imbalanced datasets, in which case we can use precision and recall

Precision

the precision of a model tells us how many of the positive predictions turned out to be correct.

Precision = TP / (TP + FP)

The better the precision, the fewer false positives we have.

Recall

Recall is the fraction of correctly classified positive examples among all positive examples

Recall = TP / (TP + FN)

the better the recall, the fewer false negatives we have.

ROC curve and AUC score

ROC - Receiver Operating Characteristic

shows how well a model can separate two classes, positive and negative

two metrics are needed for ROC curves:

- TPR (True positive rate)
  - TPR = TP / (TP + FN)
- FPR (False positive rate)
  - FPR = TP / (TP + FN)

In [74]:
```
scores = []

thresholds = np.linspace(0, 1, 101)

for t in thresholds:
    tp = ((y_pred >= t) & (y_val == 1)).sum()
    fp = ((y_pred >= t) & (y_val == 0)).sum()
    fn = ((y_pred < t) & (y_val == 1)).sum()
    tn = ((y_pred < t) & (y_val == 0)).sum()
    scores.append((t, tp, fp, fn, tn))
```

In [75]:
```
df_scores = pd.DataFrame(scores)
df_scores.columns = ['threshold', 'tp', 'fp', 'fn', 'tn']
```

change the list into a pandas dataframe (b/c df are easier to handle than a list of tuples

In [76]:
```
df_scores['tpr'] = df_scores.tp / (df_scores.tp + df_scores.fn)
df_scores['fpr'] = df_scores.fp / (df_scores.fp + df_scores.tn)
```

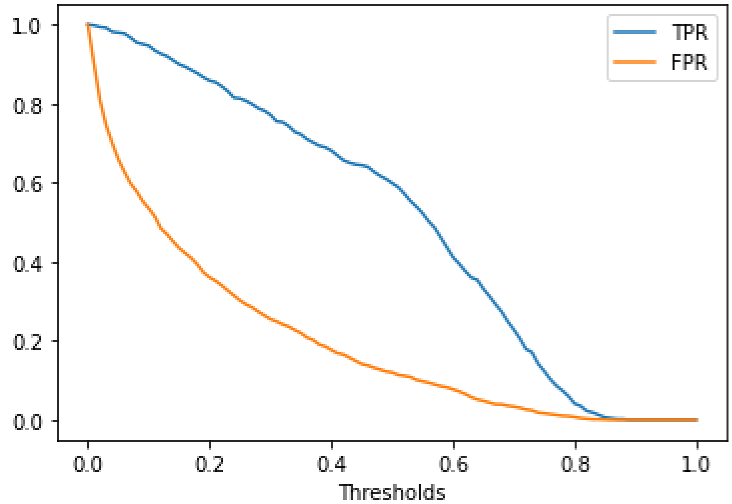
In [77]:
```
df_scores[::10]
```

Out[77]:

	threshold	tp	fp	fn	tn	tpr	fpr
0	0.0	486	1374	0	0	1.000000	1.000000
10	0.1	460	735	26	639	0.946502	0.534934
20	0.2	417	495	69	879	0.858025	0.360262
30	0.3	375	350	111	1024	0.771605	0.254731
40	0.4	331	243	155	1131	0.681070	0.176856
50	0.5	291	165	195	1209	0.598765	0.120087
60	0.6	200	106	286	1268	0.411523	0.077147
70	0.7	111	46	375	1328	0.228395	0.033479
80	0.8	20	10	466	1364	0.041152	0.007278
90	0.9	0	0	486	1374	0.000000	0.000000
100	1.0	0	0	486	1374	0.000000	0.000000

In [78]:
```
plt.plot(df_scores.threshold, df_scores.tpr, label="TPR")
plt.plot(df_scores.threshold, df_scores.fpr, label='FPR')
plt.xlabel('Thresholds')
plt.legend()
```

Out[78]: <matplotlib.legend.Legend at 0x7f90a51ba3a0>



As the threshold grows, both metrics decline but at different rates..

Ideally, FPR should go down very quickly. A small FPR indicates that the model makes very few mistakes predicting negative examples (false positives).

To better understand what these TPR and FPR mean, we'll compare them with two baseline models: a random model and the ideal model.

Random baseline model

A random model outputs a random score between 0 and 1, regardless of the input.

In [79]:
```
np.random.seed(1)
y_rand = np.random.uniform(0, 1, size=len(y_val))
```

In [80]:
```
def tpr_fpr_dataframe(y_val, y_pred):
    scores = []

    thresholds = np.linspace(0, 1, 101)

    for t in thresholds:
        tp = ((y_pred >= t) & (y_val == 1)).sum()
        fp = ((y_pred >= t) & (y_val == 0)).sum()
        fn = ((y_pred < t) & (y_val == 1)).sum()
        tn = ((y_pred < t) & (y_val == 0)).sum()

        scores.append((t, tp, fp, fn, tn))

    df_scores = pd.DataFrame(scores)
    df_scores.columns = ['threshold', 'tp', 'fp', 'fn', 'tn']

    df_scores['tpr'] = df_scores.tp / (df_scores.tp + df_scores.fn)
    df_scores['fpr'] = df_scores.fp / (df_scores.fp + df_scores.tn)

    return df_scores
```

we'll use the above function to calculate the TPR and FPR for the random model:

In [81]:
```
df_rand = tpr_fpr_dataframe(y_val, y_rand)
```

now we create a datagrame with TPR and FPR values at different thresholds

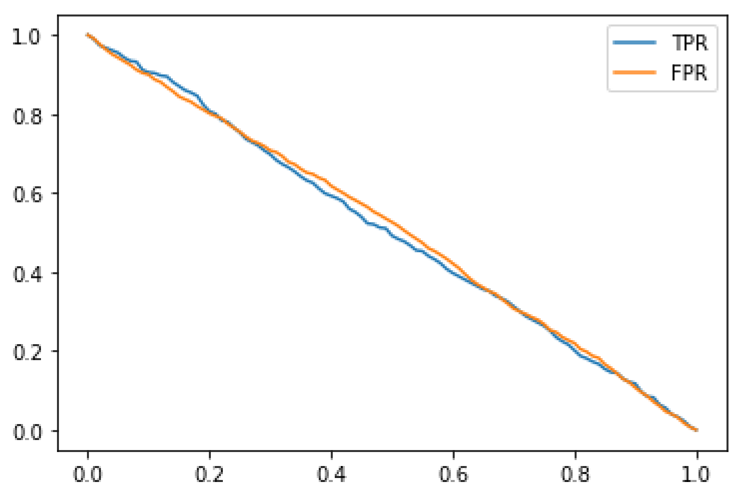
In [82]:
```
np.random.seed(1)
y_rand = np.random.uniform(0, 1, size=len(y_val))
df_rand = tpr_fpr_dataframe(y_val, y_rand)
df_rand[::10]
```

Out[82]:

	threshold	tp	fp	fn	tn	tpr	fpr
0	0.0	486	1374	0	0	1.000000	1.000000
10	0.1	440	1236	46	138	0.905350	0.899563
20	0.2	392	1101	94	273	0.806584	0.801310
30	0.3	339	972	147	402	0.697531	0.707424
40	0.4	288	849	198	525	0.592593	0.617904
50	0.5	239	723	247	651	0.491770	0.526201
60	0.6	193	579	293	795	0.397119	0.421397
70	0.7	152	422	334	952	0.312757	0.307132
80	0.8	98	302	388	1072	0.201646	0.219796
90	0.9	57	147	429	1227	0.117284	0.106987
100	1.0	0	0	486	1374	0.000000	0.000000

In [83]:
```
plt.plot(df_rand.threshold, df_rand.tpr, label='TPR')
plt.plot(df_rand.threshold, df_rand.fpr, label='FPR')
plt.legend()
```

Out[83]: <matplotlib.legend.Legend at 0x7f9081d00610>



Ideal Model

The ideal model always makes correct decisions.

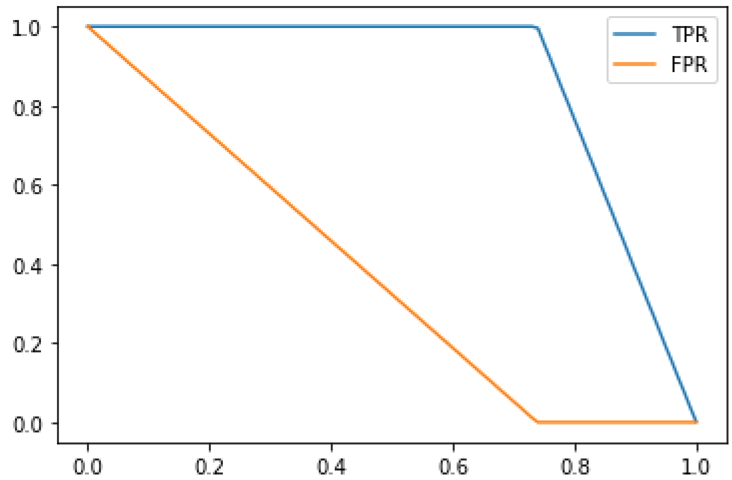
In [84]:
```
num_neg = (y_val == 0).sum()
num_pos = (y_val == 1).sum()

y_ideal = np.repeat([0, 1], [num_neg, num_pos])
y_pred_ideal = np.linspace(0, 1, num_neg + num_pos)

df_ideal = tpr_fpr_dataframe(y_ideal, y_pred_ideal)
```

In [85]:
```
plt.plot(df_ideal.threshold, df_ideal.tpr, label='TPR')
plt.plot(df_ideal.threshold, df_ideal.fpr, label='FPR')
plt.legend()
```

Out[85]:
```
<matplotlib.legend.Legend at 0x7f9082daed30>
```



ROC Curve

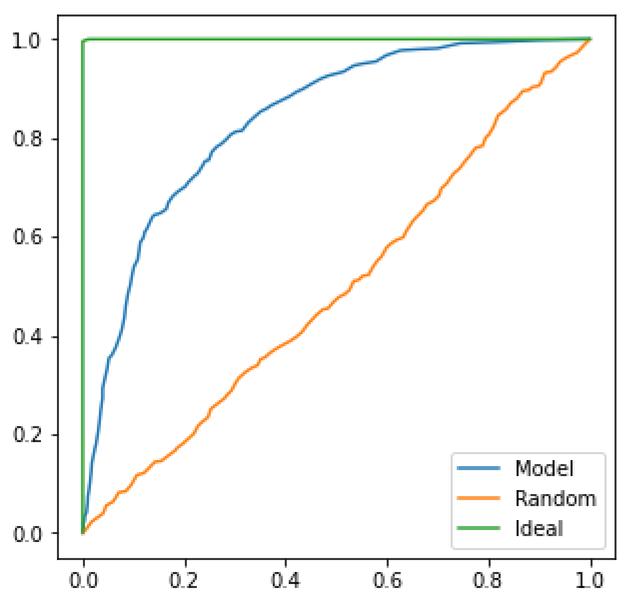
To create an ROC curve, instead of plotting FPR and TPR against different threshold values, we plot them against each other.

In [86]:
```
plt.figure(figsize=(5, 5))

plt.plot(df_scores.fpr, df_scores.tpr, label='Model')
plt.plot(df_rand.fpr, df_rand.tpr, label='Random')
plt.plot(df_ideal.fpr, df_ideal.tpr, label='Ideal')

plt.legend()
```

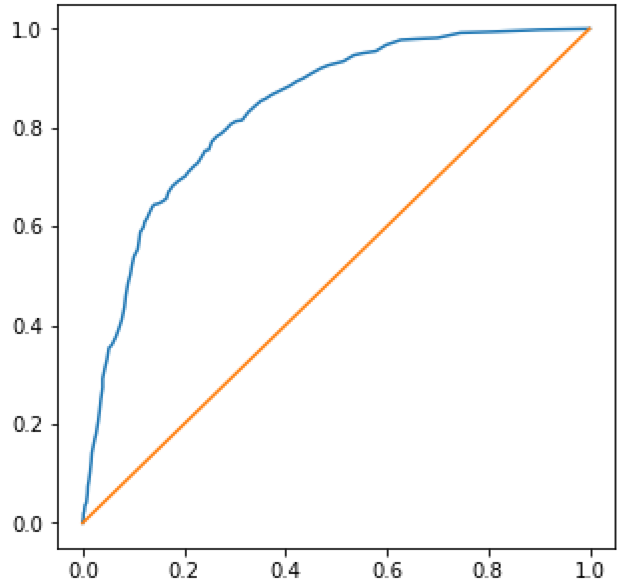
Out[86]:
```
<matplotlib.legend.Legend at 0x7f9081d7c1c0>
```



We want our model to be as close to the ideal curve as possible and as far from the random curve

In [87]:
```
plt.figure(figsize=(5, 5))
plt.plot(df_scores.fpr, df_scores.tpr)
plt.plot([0, 1], [0, 1])
```

Out[87]: [<matplotlib.lines.Line2D at 0x7f90830c2130>]

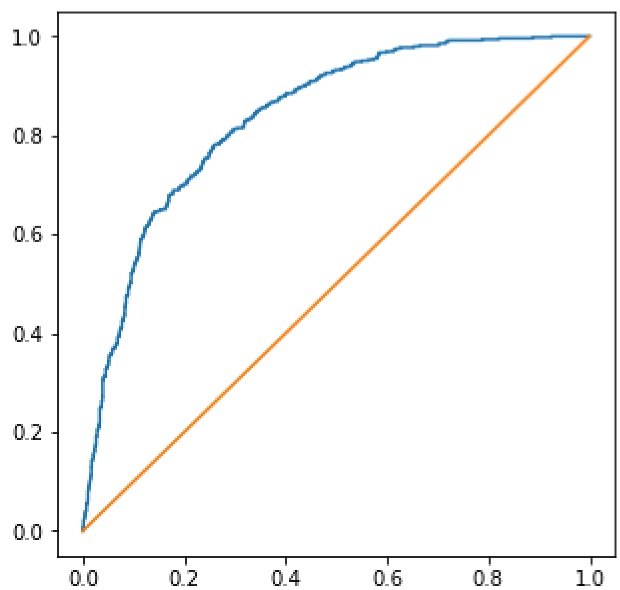


roc_curve function from the metrics package of scikit-learn

In [88]:
```
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(y_val, y_pred)

plt.figure(figsize=(5, 5))
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1])
```

Out[88]: [<matplotlib.lines.Line2D at 0x7f9083149a00>]



Exmple of the ROC curves of teh large and small models and plot them on the same graph

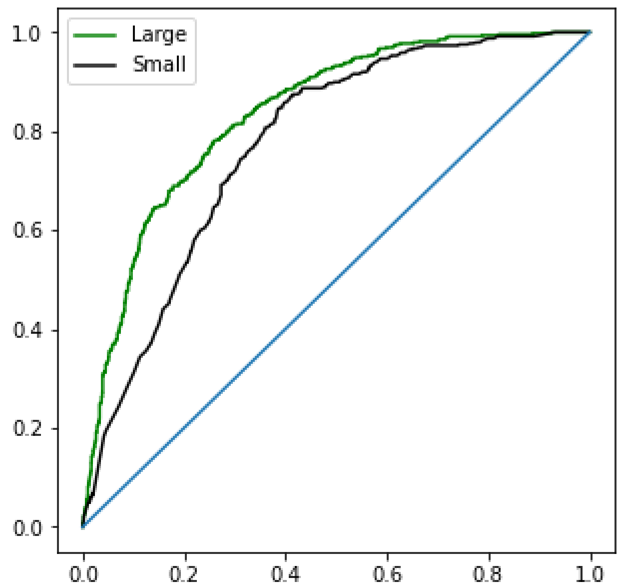
In [91]:
```
fpr_large, tpr_large, _ = roc_curve(y_val, y_pred)
fpr_small, tpr_small, _ = roc_curve(y_val, y_pred_small)

plt.figure(figsize=(5, 5))

plt.plot(fpr_large, tpr_large, color='green', label='Large')
plt.plot(fpr_small, tpr_small, color='black', label='Small')
plt.plot([0, 1], [0, 1])
plt.legend()
```

<matplotlib.legend.Legend at 0x7f90832bcc40>

Out[91]:



Area under the ROC curve (AUC)

auc is a metric for evaluating the performance of a binary classification model.

used to quantify the closeness, by measuring the are under the ROC curve

An AUC of 0.9 is indicative of a reasonably good model; 0.8 is okay, 0.7 is not very performant, and 0.6 indicates quite poor performance

In [92]:
```
from sklearn.metrics import auc
auc(df_scores.fpr, df_scores.tpr)
```

Out[92]: 0.8363905511528025

In [93]:
```
df_scores_small = tpr_fpr_dataframe(y_val, y_pred_small)
```

In [94]:
```
auc(df_scores_small.fpr, df_scores_small.tpr)
```

Out[94]: 0.7732335076464139

In [95]:
```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_val, y_pred)
```

Out[95]: 0.8366990433746054

ROC curveas and AUC scores tell us how well the model separates positive and negative examples

In [96]:
```
roc_auc_score(y_val, y_pred_small)
```

Out[96]: 0.7746531708807303

Interpretation of AUC: the probability that a randomly chosen positive example ranks higher than a randomly chosen negative example

In [97]:
```
neg = y_pred[y_val == 0]
pos = y_pred[y_val == 1]
```

```
np.random.seed(1)
neg_choice = np.random.randint(low=0, high=len(neg), size=10000)
pos_choice = np.random.randint(low=0, high=len(pos), size=10000)
(pos[pos_choice] > neg[neg_choice]).mean()
```

Out[97]:

```
0.8368
```

Parameter Tuning

K-fold cross-validation

first we'll put all the code for training into train function, which first converts the data into a one-hot encoding representation and then trains the model

In [98]:

```
def train(df, y):
    cat = df[categorical + numerical].to_dict(orient='records')

    dv = DictVectorizer(sparse=False)
    dv.fit(cat)

    X = dv.transform(cat)

    model = LogisticRegression(solver='liblinear')
    model.fit(X, y)

    return dv, model
```

We also put the prediction logic into a predict funtion, which takes in a dagtaframe with customers, the vectorizer we "trained" previously - for doing some one hot encoding - and the model. Then we apply the vectorizer to the dataframe, get a matrix, and finally apply the model to the matrix to get predictions.

In [99]:

```
def predict(df, dv, model):
    cat = df[categorical + numerical].to_dict(orient='records')

    X = dv.transform(cat)
    y_pred = model.predict_proba(X)[:, 1]

    return y_pred
```

Now we can use these functions for implementing K-fold cross-validation.

We don't need to implement cross-validation ourselves: in Scikit-learn there's a class for doing that. It's called KFold and it lives in the model_selection package

In [100…

```
from sklearn.model_selection import KFold

kfold = KFold(n_splits=10, shuffle=True, random_state=1)

aucs = []

for train_idx, val_idx in kfold.split(df_train_full):
    df_train = df_train_full.iloc[train_idx]
    y_train = df_train.churn.values

    df_val = df_train_full.iloc[val_idx]
    y_val = df_val.churn.values
```

```
    dv, model = train(df_train, y_train)
    y_pred = predict(df_val, dv, model)

    rocauc = roc_auc_score(y_val, y_pred)
    aucs.append(rocauc)
```

In [101…

```
np.array(aucs)
```

Out[101]:

```
array([0.85185185, 0.84178527, 0.85651878, 0.82892766, 0.82703164,
       0.83812854, 0.83871703, 0.82771189, 0.84256707, 0.85679473])
```

In [102…

```
print('auc = %0.3f ± %0.3f' % (np.mean(aucs), np.std(aucs)))
```

```
auc = 0.841 ± 0.011
```

Now, not only do we know the average performance, but we also have an idea of how volatile that performance is, or how far it may deviate from the average

Now we can use K-fold cross-validation for parameter tuning (selecting the best parameters)

below we adjust the train function to take in an additional parameter

In [111…

```
def train(df, y, C):
    cat = df[categorical + numerical].to_dict(orient='records')

    dv = DictVectorizer(sparse=False)
    dv.fit(cat)

    X = dv.transform(cat)

    model = LogisticRegression(solver='liblinear', C=C)
    model.fit(X, y)

    return dv, model
```

Nowewe'll find the best parameter C.

- Loop over different values of C.
- For each C, run cross-validation and record the mean AUC across all folds as well as the standard deviation

In [112…

```
nfolds = 5
kfold = KFold(n_splits=nfolds, shuffle=True, random_state=1)

for C in [0.001, 0.01, 0.1, 0.5, 1, 10]:
    aucs = []

    for train_idx, val_idx in kfold.split(df_train_full):
        df_train = df_train_full.iloc[train_idx]
        df_val = df_train_full.iloc[val_idx]

        y_train = df_train.churn.values
        y_val = df_val.churn.values

        dv, model = train(df_train, y_train, C=C)
        y_pred = predict(df_val, dv, model)

        auc = roc_auc_score(y_val, y_pred)
        aucs.append(auc)

    print('C=%s, auc = %0.3f ± %0.3f' % (C, np.mean(aucs), np.std(aucs)))
```

```
C=0.001, auc = 0.819 ± 0.013
```

```
C=0.01, auc = 0.838 ± 0.008
C=0.1, auc = 0.840 ± 0.007
C=0.5, auc = 0.840 ± 0.007
C=1, auc = 0.840 ± 0.007
C=10, auc = 0.835 ± 0.006
```

When the C parameter is small, the model is more regularized. The weights of the model are more restricted, so in general, rhey are smaller. Small weights in the model give us additional assurance that the model will behave well when we use it on real data.

In [118…

```
y_train = df_train_full.churn.values
y_test = df_test.churn.values

dv, model = train(df_train_full, y_train, C=0.5)
y_pred = predict(df_test, dv, model)

auc = roc_auc_score(y_test, y_pred)
print('auc = %.3f' % auc)
```

```
auc = 0.857
```