

ex3

July 19, 2024

```
[ ]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score
```

```
[ ]: df = pd.read_csv('datasets\churn-bigml-20.csv')
df.head()
```

```
[ ]: State Account length Area code International plan Voice mail plan \
0 LA 117 408 No No
1 IN 65 415 No No
2 NY 161 415 No No
3 SC 111 415 No No
4 HI 49 510 No No

Number vmail messages Total day minutes Total day calls \
0 0 184.5 97
1 0 129.1 137
2 0 332.9 67
3 0 110.4 103
4 0 119.3 117

Total day charge Total eve minutes Total eve calls Total eve charge \
0 31.37 351.6 80 29.89
1 21.95 228.5 83 19.42
2 56.59 317.8 97 27.01
3 18.77 137.3 102 11.67
4 20.28 215.1 109 18.28

Total night minutes Total night calls Total night charge \
0 215.8 90 9.71
1 208.8 111 9.40
2 160.6 128 7.23
3 189.6 105 8.53
4 178.7 90 8.04

Total intl minutes Total intl calls Total intl charge \
```

0	8.7	4	2.35
1	12.7	6	3.43
2	5.4	9	1.46
3	7.7	6	2.08
4	11.1	1	3.00

	Customer service calls	Churn
0	1	False
1	4	True
2	4	True
3	2	False
4	1	False

```
[ ]: df.shape
```

```
[ ]: (667, 20)
```

```
[ ]: import seaborn as sns
```

```
#sns.pairplot(df)
```

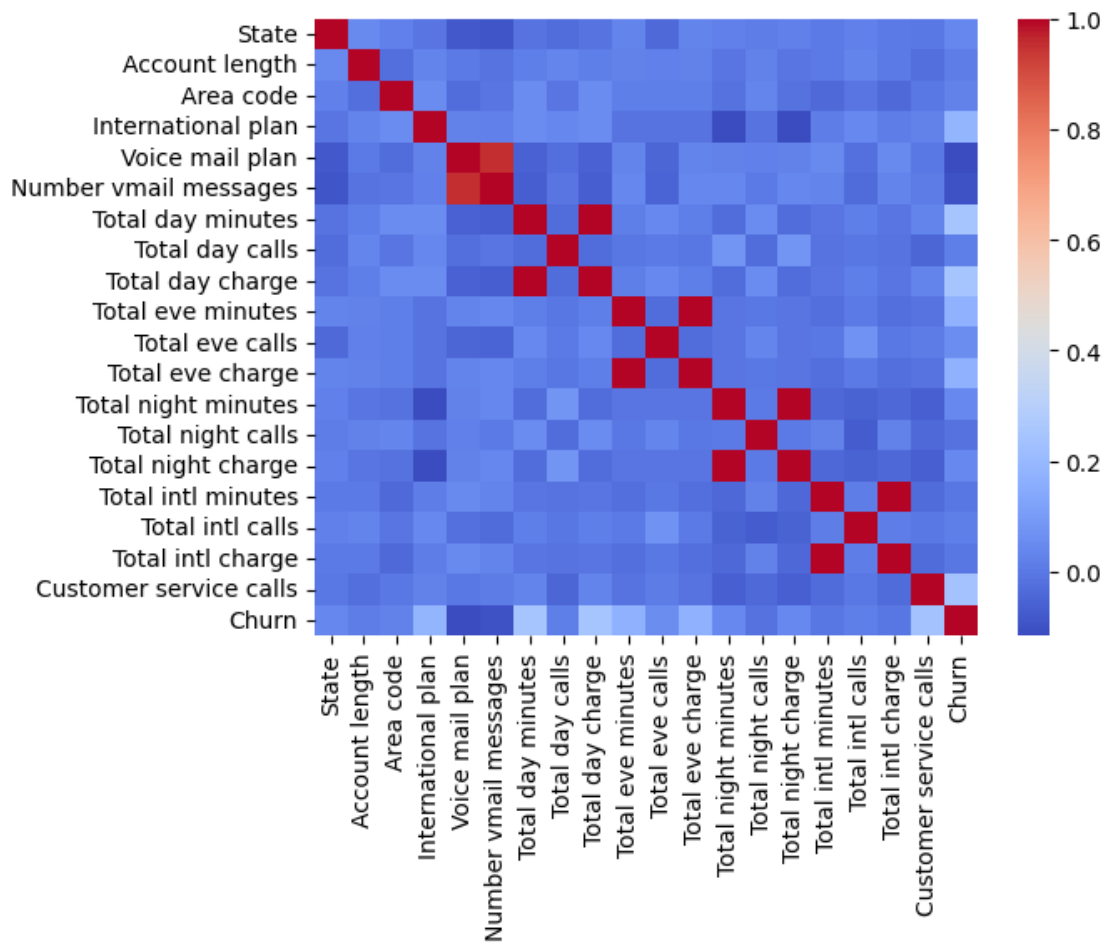
```
[ ]: model = LogisticRegression(max_iter=1580)
le = LabelEncoder()

df['International plan'] = le.fit_transform(df['International plan'])
df['Voice mail plan'] = le.fit_transform(df['Voice mail plan'])
df['Churn'] = le.fit_transform(df['Churn'])
df['State'] = le.fit_transform(df['State'])

X = df.drop('Churn', axis=1)
y = df['Churn']
```

```
[ ]: sns.heatmap(df.corr(), cmap='coolwarm')
```

```
[ ]: <Axes: >
```



```
[ ]: #After Pre-processing
```

```
X.head()
```

```
[ ]:   State  Account length  Area code  International plan  Voice mail plan  \
0     18             117         408                   0             0
1     15              65         415                   0             0
2     34             161         415                   0             0
3     40             111         415                   0             0
4     11              49         510                   0             0

   Number vmail messages  Total day minutes  Total day calls  \
0                      0             184.5             97
1                      0             129.1            137
2                      0             332.9             67
3                      0             110.4            103
4                      0             119.3            117
```

	Total day charge	Total eve minutes	Total eve calls	Total eve charge \
0	31.37	351.6	80	29.89
1	21.95	228.5	83	19.42
2	56.59	317.8	97	27.01
3	18.77	137.3	102	11.67
4	20.28	215.1	109	18.28

	Total night minutes	Total night calls	Total night charge \
0	215.8	90	9.71
1	208.8	111	9.40
2	160.6	128	7.23
3	189.6	105	8.53
4	178.7	90	8.04

	Total intl minutes	Total intl calls	Total intl charge \
0	8.7	4	2.35
1	12.7	6	3.43
2	5.4	9	1.46
3	7.7	6	2.08
4	11.1	1	3.00

	Customer service calls
0	1
1	4
2	4
3	2
4	1

```
[ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
↳ random_state=42)
```

```
model.fit(X_train, y_train)
print(f"Accuracy :{model.score(X_test, y_test)}")
```

Accuracy :0.8905472636815921

```
[ ]: y_pred = model.predict(X_test)
y_pred
```

```
[ ]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0], dtype=int64)
```

```
[ ]: import numpy as np

individual_values = np.array([18,117,408,0,0,0,184.5,97,31.37,351.6,80,29.
↪89,215.8,90,9.71,8.7,4,2.35,1]).reshape(1,- 1) # Select the rows you want
↪to predict
predictions = model.predict(individual_values)
print(predictions)
```

```
[0]
```

```
c:\Python311\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have
valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(
```

```
[ ]: model.predict_proba(X_test)
```

```
[ ]: array([[0.96707203, 0.03292797],
[0.96080303, 0.03919697],
[0.93471187, 0.06528813],
[0.66630954, 0.33369046],
[0.67402722, 0.32597278],
[0.96626244, 0.03373756],
[0.96255959, 0.03744041],
[0.91412764, 0.08587236],
[0.84048346, 0.15951654],
[0.99498264, 0.00501736],
[0.78172667, 0.21827333],
[0.94393493, 0.05606507],
[0.8503402 , 0.1496598 ],
[0.97745884, 0.02254116],
[0.9631015 , 0.0368985 ],
[0.99628937, 0.00371063],
[0.82309954, 0.17690046],
[0.97984409, 0.02015591],
[0.55730758, 0.44269242],
[0.9817523 , 0.0182477 ],
[0.33137369, 0.66862631],
[0.66907177, 0.33092823],
[0.8975433 , 0.1024567 ],
[0.4820346 , 0.5179654 ],
[0.96956726, 0.03043274],
[0.85279512, 0.14720488],
[0.9403649 , 0.0596351 ],
[0.99360996, 0.00639004],
[0.91731378, 0.08268622],
```

[0.98015563, 0.01984437],
[0.9815212 , 0.0184788],
[0.80640618, 0.19359382],
[0.44352059, 0.55647941],
[0.93378629, 0.06621371],
[0.99130625, 0.00869375],
[0.89982472, 0.10017528],
[0.95205479, 0.04794521],
[0.90854844, 0.09145156],
[0.95350962, 0.04649038],
[0.99709527, 0.00290473],
[0.83113797, 0.16886203],
[0.95841055, 0.04158945],
[0.90486064, 0.09513936],
[0.91336882, 0.08663118],
[0.9661468 , 0.0338532],
[0.97155784, 0.02844216],
[0.74282678, 0.25717322],
[0.97559381, 0.02440619],
[0.94955985, 0.05044015],
[0.89986598, 0.10013402],
[0.88864625, 0.11135375],
[0.89105207, 0.10894793],
[0.81702026, 0.18297974],
[0.89644707, 0.10355293],
[0.88423092, 0.11576908],
[0.83773203, 0.16226797],
[0.95427111, 0.04572889],
[0.76553494, 0.23446506],
[0.93883389, 0.06116611],
[0.95698529, 0.04301471],
[0.96891171, 0.03108829],
[0.9972614 , 0.0027386],
[0.08863486, 0.91136514],
[0.37690393, 0.62309607],
[0.77768132, 0.22231868],
[0.97316805, 0.02683195],
[0.82114375, 0.17885625],
[0.91336204, 0.08663796],
[0.33568512, 0.66431488],
[0.93530973, 0.06469027],
[0.9768114 , 0.0231886],
[0.81319104, 0.18680896],
[0.82689204, 0.17310796],
[0.92420732, 0.07579268],
[0.86755417, 0.13244583],
[0.97559912, 0.02440088],

[0.40893102, 0.59106898],
[0.94536738, 0.05463262],
[0.99068815, 0.00931185],
[0.88057717, 0.11942283],
[0.7798378 , 0.2201622],
[0.9546383 , 0.0453617],
[0.9883814 , 0.0116186],
[0.40743361, 0.59256639],
[0.86443433, 0.13556567],
[0.72845197, 0.27154803],
[0.96858489, 0.03141511],
[0.93011413, 0.06988587],
[0.85707041, 0.14292959],
[0.82763557, 0.17236443],
[0.7924638 , 0.2075362],
[0.80021676, 0.19978324],
[0.23982052, 0.76017948],
[0.98702488, 0.01297512],
[0.89667684, 0.10332316],
[0.93720415, 0.06279585],
[0.97811323, 0.02188677],
[0.9703831 , 0.0296169],
[0.91673961, 0.08326039],
[0.9339233 , 0.0660767],
[0.95092626, 0.04907374],
[0.61133898, 0.38866102],
[0.93079973, 0.06920027],
[0.93464211, 0.06535789],
[0.91128289, 0.08871711],
[0.98864331, 0.01135669],
[0.92310281, 0.07689719],
[0.84622553, 0.15377447],
[0.77632415, 0.22367585],
[0.98084167, 0.01915833],
[0.60380493, 0.39619507],
[0.42064041, 0.57935959],
[0.95279733, 0.04720267],
[0.96808406, 0.03191594],
[0.86940848, 0.13059152],
[0.82638007, 0.17361993],
[0.75504734, 0.24495266],
[0.91065777, 0.08934223],
[0.95119412, 0.04880588],
[0.97497736, 0.02502264],
[0.97496925, 0.02503075],
[0.82994257, 0.17005743],
[0.66677224, 0.33322776],

[0.96042965, 0.03957035],
[0.94307499, 0.05692501],
[0.78855719, 0.21144281],
[0.94421442, 0.05578558],
[0.97489002, 0.02510998],
[0.97620533, 0.02379467],
[0.97434349, 0.02565651],
[0.98020099, 0.01979901],
[0.99180817, 0.00819183],
[0.98247297, 0.01752703],
[0.93733043, 0.06266957],
[0.97133606, 0.02866394],
[0.85799219, 0.14200781],
[0.96343942, 0.03656058],
[0.69118487, 0.30881513],
[0.6056906 , 0.3943094],
[0.96442305, 0.03557695],
[0.97310928, 0.02689072],
[0.88549843, 0.11450157],
[0.93755084, 0.06244916],
[0.76664215, 0.23335785],
[0.94965667, 0.05034333],
[0.87773515, 0.12226485],
[0.83521871, 0.16478129],
[0.85144365, 0.14855635],
[0.97138911, 0.02861089],
[0.96822924, 0.03177076],
[0.95886015, 0.04113985],
[0.69084495, 0.30915505],
[0.98462292, 0.01537708],
[0.77163828, 0.22836172],
[0.98816954, 0.01183046],
[0.72075861, 0.27924139],
[0.96664713, 0.03335287],
[0.92040342, 0.07959658],
[0.89684158, 0.10315842],
[0.77864826, 0.22135174],
[0.95449941, 0.04550059],
[0.66436333, 0.33563667],
[0.66759931, 0.33240069],
[0.95897745, 0.04102255],
[0.19283776, 0.80716224],
[0.57995602, 0.42004398],
[0.6873538 , 0.3126462],
[0.67548875, 0.32451125],
[0.68810474, 0.31189526],
[0.7981786 , 0.2018214],

[0.94052002, 0.05947998],
[0.92863047, 0.07136953],
[0.89929792, 0.10070208],
[0.41636188, 0.58363812],
[0.96171914, 0.03828086],
[0.89274934, 0.10725066],
[0.7485307 , 0.2514693],
[0.71430692, 0.28569308],
[0.99516069, 0.00483931],
[0.97874772, 0.02125228],
[0.83854007, 0.16145993],
[0.96883681, 0.03116319],
[0.96237742, 0.03762258],
[0.52776644, 0.47223356],
[0.92470819, 0.07529181],
[0.946859 , 0.053141],
[0.85879293, 0.14120707],
[0.34112668, 0.65887332],
[0.96938658, 0.03061342],
[0.90841739, 0.09158261],
[0.89381327, 0.10618673],
[0.94397018, 0.05602982],
[0.8958914 , 0.1041086],
[0.33745723, 0.66254277],
[0.94353863, 0.05646137],
[0.98798281, 0.01201719],
[0.5210601 , 0.4789399],
[0.87028137, 0.12971863],
[0.65221205, 0.34778795],
[0.94922726, 0.05077274],
[0.96232678, 0.03767322]])