

```
In [1]: import pandas as pd
data = pd.read_csv('bigml_59c28831336c6604c800002a.csv')
data.head()
```

Out[1]:

	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	...	total eve calls	total eve charge	total night minutes	total night calls	total night charge	total intl minutes	total intl calls	c
0	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	...	99	16.78	244.7	91	11.01	10.0	3	
1	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	...	103	16.62	254.4	103	11.45	13.7	3	
2	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	...	110	10.30	162.6	104	7.32	12.2	5	
3	OH	84	408	375-9999	yes	no	0	299.4	71	50.90	...	88	5.26	196.9	89	8.86	6.6	7	
4	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	...	122	12.61	186.9	121	8.41	10.1	3	

5 rows × 21 columns

Import sklearn model selection train test split module

```
In [6]: from sklearn.model_selection import train_test_split

y = data['churn']
X = data.drop(['state', 'account length', 'area code', 'phone number', 'international plan', 'voice mail plan', 'churn'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

```
In [16]: from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
import numpy as np

baseline_model = LogisticRegression(random_state=42)

baseline_neg_log_loss_cv = cross_val_score(baseline_model, X_train, y_train, cv=3, scoring="neg_log_loss")

baseline_log_loss = -(baseline_neg_log_loss_cv.mean())
baseline_log_loss
```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

Out[16]: 0.37897883384972086

```
In [17]: baseline_model = LogisticRegression(random_state=42)
baseline_neg_log_loss_cv = cross_val_score(baseline_model, X_train, y_train, scoring="neg_log_loss")
baseline_log_loss = -(baseline_neg_log_loss_cv.mean())
baseline_log_loss
```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(  

```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(  

```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(  

```

```
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

```
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
n_iter_i = _check_optimize_result(
C:\Users\kuta\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

```
https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
```

Please also refer to the documentation for alternative solver options:

```
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
```

```
n_iter_i = _check_optimize_result(
```

Out[17]: 0.3778957231977595

```

In [24]: from sklearn.metrics import make_scorer
        from sklearn.model_selection import StratifiedKFold
        from sklearn.base import clone

        neg_log_loss = make_scorer(log_loss, greater_is_better=False, needs_proba=True)

        baseline_model = LogisticRegression(random_state=42)

        kfold_scores = np.ndarray(5)

        kfold = StratifiedKFold()
        for fold, (train_index, val_index) in enumerate(kfold.split(X_train, y_train)):
            X_t, X_val = X_train.iloc[train_index], X_train.iloc[val_index]
            y_t, y_val = y_train.iloc[train_index], y_train.iloc[val_index]

            temp_model = clone(baseline_model)
            temp_model.fit(X_t, y_t)

            neg_log_loss_score = neg_log_loss(temp_model, X_val, y_val)
            kfold_scores[fold] = neg_log_loss_score

        -(kfold_scores.mean())

```

```

-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\4196509598.py in <module>
      3 from sklearn.base import clone
      4
----> 5 neg_log_loss = make_scorer(log_loss, greater_is_better=False, needs_proba=True)
      6
      7 baseline_model = LogisticRegression(random_state=42)

NameError: name 'log_loss' is not defined

```

```

In [31]: from sklearn.base import clone
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE

def custom_cross_val_score(estimator, X, y):

    kfold_train_scores = np.ndarray(5)
    kfold_val_scores = np.ndarray(5)

    kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
    for fold, (train_index, val_index) in enumerate(kfold.split(X, y)):
        X_t, X_val = X.iloc[train_index], X.iloc[val_index]
        y_t, y_val = y.iloc[train_index], y.iloc[val_index]

        scaler = StandardScaler()
        X_t_scaled = scaler.fit_transform(X_t)
        X_val_scaled = scaler.transform(X_val)

        sm = SMOTE(sampling_strategy=0.28, random_state=42)
        X_t_oversampled, y_t_oversampled = sm.fit_resample(X_t_scaled, y_t)

        temp_model = clone(estimator)
        temp_model.fit(X_t_oversampled, y_t_oversampled)

        neg_log_loss_score_train = -log_loss(y_t_oversampled, temp_model.predict_proba(X_t_oversampled))
        neg_log_loss_score_val = -log_loss(y_val, temp_model.predict_proba(X_val_scaled))
        kfold_train_scores[fold] = neg_log_loss_score_train
        kfold_val_scores[fold] = neg_log_loss_score_val

    return kfold_train_scores, kfold_val_scores

model_with_preprocessing = LogisticRegression(random_state=42, class_weight={1: 0.28})
preprocessed_train_scores, preprocessed_neg_log_loss_cv = custom_cross_val_score(model_with_preprocessing, X_train, y_train)
-(preprocessed_neg_log_loss_cv.mean())

```



```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\241692327.py in <module>
    32
    33 model_with_preprocessing = LogisticRegression(random_state=42, class_weight={1: 0.28})
----> 34 preprocessed_train_scores, preprocessed_neg_log_loss_cv = custom_cross_val_score(model_with_preprocessing, X_train, y_train)
    35 -(preprocessed_neg_log_loss_cv.mean())

~\AppData\Local\Temp\ipykernel_28304\241692327.py in custom_cross_val_score(estimator, X, y)
    24     temp_model.fit(X_t_oversampled, y_t_oversampled)
    25
----> 26     neg_log_loss_score_train = -log_loss(y_t_oversampled, temp_model.predict_proba(X_t_oversampled))
    27     neg_log_loss_score_val = -log_loss(y_val, temp_model.predict_proba(X_val_scaled))
    28     kfold_train_scores[fold] = neg_log_loss_score_train

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

```

In [40]: from sklearn.base import clone
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from sklearn.metrics import log_loss # Add this import

def custom_cross_val_score(estimator, X, y):

    kfold_train_scores = np.ndarray(5)
    kfold_val_scores = np.ndarray(5)

    kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
    for fold, (train_index, val_index) in enumerate(kfold.split(X, y)):
        X_t, X_val = X.iloc[train_index], X.iloc[val_index]
        y_t, y_val = y.iloc[train_index], y.iloc[val_index]

        scaler = StandardScaler()
        X_t_scaled = scaler.fit_transform(X_t)
        X_val_scaled = scaler.transform(X_val)

        sm = SMOTE(sampling_strategy=0.28, random_state=42)
        X_t_oversampled, y_t_oversampled = sm.fit_resample(X_t_scaled, y_t)

        temp_model = clone(estimator)
        temp_model.fit(X_t_oversampled, y_t_oversampled)

        neg_log_loss_score_train = -log_loss(y_t_oversampled, temp_model.predict_proba(X_t_oversampled))
        neg_log_loss_score_val = -log_loss(y_val, temp_model.predict_proba(X_val_scaled))
        kfold_train_scores[fold] = neg_log_loss_score_train
        kfold_val_scores[fold] = neg_log_loss_score_val

    return kfold_train_scores, kfold_val_scores

model_with_preprocessing = LogisticRegression(random_state=42, class_weight={1: 0.28})
preprocessed_train_scores, preprocessed_neg_log_loss_cv = custom_cross_val_score(model_with_preprocessing, X_train, y_train)
-(preprocessed_neg_log_loss_cv.mean())

```

Out[40]: 0.3853795144663737

```
In [41]: print(-baseline_neg_log_loss_cv.mean())
print(-preprocessed_neg_log_loss_cv.mean())
```

```
0.3778957231977595
0.3853795144663737
```

```
In [ ]:
```

```
In [44]: print("Previous Model")
print("Train average:      ", -preprocessed_train_scores.mean())
print("Validation average:", -preprocessed_neg_log_loss_cv.mean())

print("Current Model")
print("Train average:      ", -less_regularization_train_scores.mean())
print("Validation average:", -less_regularization_val_scores.mean())
```

```
Previous Model
Train average:      0.5338333518339983
Validation average: 0.3853795144663737
Current Model
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\1234242582.py in <module>
      4
      5 print("Current Model")
----> 6 print("Train average:      ", -less_regularization_train_scores.mean())
      7 print("Validation average:", -less_regularization_val_scores.mean())

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

```
In [47]: print("solver:", model_less_regularization.get_params()["solver"])
print("penalty:", model_less_regularization.get_params()["penalty"])
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\2640628920.py in <module>
----> 1 print("solver:", model_less_regularization.get_params()["solver"])
      2 print("penalty:", model_less_regularization.get_params()["penalty"])

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

```
In [48]: model_more_iterations = LogisticRegression(random_state=42, class_weight={1: 0.28}, max_iter=1000) # Define the model with more iterations

more_iterations_train_scores, more_iterations_val_scores = custom_cross_val_score(
    model_more_iterations,
    X_train,
    y_train
)

print("Previous Best Model (Less Regularization)")
print("Train average:      ", -less_regularization_train_scores.mean())
print("Validation average:", -less_regularization_val_scores.mean())
print("Previous Model with This Solver")
print("Train average:      ", -alternative_solver_train_scores.mean())
print("Validation average:", -alternative_solver_val_scores.mean())
print("Current Model (More Iterations)")
print("Train average:      ", -more_iterations_train_scores.mean())
print("Validation average:", -more_iterations_val_scores.mean())
```

Previous Best Model (Less Regularization)

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\1784884800.py in <module>
      8
      9 print("Previous Best Model (Less Regularization)")
--> 10 print("Train average:      ", -less_regularization_train_scores.mean())
     11 print("Validation average:", -less_regularization_val_scores.mean())
     12 print("Previous Model with This Solver")

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

```
In [45]: final_model = model_less_regularization

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

sm = SMOTE(sampling_strategy=0.28, random_state=42)

X_train_oversampled, y_train_oversampled = sm.fit_resample(X_train_scaled, y_train)
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\2854363567.py in <module>
----> 1 final_model = model_less_regularization
      2
      3 scaler = StandardScaler()
      4
      5 X_train_scaled = scaler.fit_transform(X_train)

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

```
In [50]: from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

sm = SMOTE(sampling_strategy=0.28, random_state=42)
X_train_oversampled, y_train_oversampled = sm.fit_resample(X_train_scaled, y_train)
```

```
In [52]: final_model.fit(X_train_oversampled, y_train_oversampled)
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\804850302.py in <module>
----> 1 final_model.fit(X_train_oversampled, y_train_oversampled)

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

```
In [37]: final_model.fit(X_train_oversampled, y_train_oversampled)

log_loss(y_test, final_model.predict_proba(X_test_scaled))
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\3236061679.py in <module>
----> 1 final_model.fit(X_train_oversampled, y_train_oversampled)
      2
      3 log_loss(y_test, final_model.predict_proba(X_test_scaled))

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

```
In [53]: from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, your_model.predict(X_test))
```

```
-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_28304\1323450376.py in <module>
      1 from sklearn.metrics import accuracy_score
      2
----> 3 accuracy = accuracy_score(y_test, your_model.predict(X_test))

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

In []: