In [8]:

```
### bosque aleatorio
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

df = pd.read_csv("data1.csv")

print(df.fillna(0))
```

```
      customerID  tenure  PhoneService  MultipleLines InternetService  \
0     0002-ORFBO       9             1              0             DSL
1     0003-MKNFE       9             1              1             DSL
2     0004-TLHLJ       4             1              0     Fiber optic
3     0011-IGKFF      13             1              0     Fiber optic
4     0013-EXCHZ       3             1              0     Fiber optic
...          ...     ...           ...            ...             ...
7038  9987-LUTYD      13             1              0             DSL
7039  9992-RRAMN      22             1              1     Fiber optic
7040  9992-UJOEL       2             1              0             DSL
7041  9993-LHIEB      67             1              0             DSL
7042  9995-HOTOH      63             0              2             DSL

      OnlineSecurity  OnlineBackup  DeviceProtection  TechSupport  \
0                  0             1                 0            1
1                  0             0                 0            0
2                  0             0                 1            0
3                  0             1                 1            0
4                  0             0                 0            1
...              ...           ...               ...          ...
7038               1             0                 0            1
7039               0             0                 0            0
7040               0             1                 0            0
7041               1             0                 1            1
7042               1             1                 1            0

      StreamingTV  StreamingMovies        Contract  PaperlessBilling  \
0               1                0        One year                 1
1               0                1  Month-to-month                 0
2               0                0  Month-to-month                 1
3               1                1  Month-to-month                 1
4               1                0  Month-to-month                 1
...           ...              ...             ...               ...
7038            0                0        One year                 0
7039            0                1  Month-to-month                 1
7040            0                0  Month-to-month                 1
7041            0                1        Two year                 0
7042            1                1        Two year                 0

         PaymentMethod  MonthlyCharges TotalCharges  Churn
0         Mailed check           65.60        593.3      0
1         Mailed check           59.90        542.4      0
2     Electronic check           73.90       280.85      1
3     Electronic check           98.00      1237.85      1
4         Mailed check           83.90        267.4      1
...                ...             ...          ...    ...
7038      Mailed check           55.15        742.9      0
7039  Electronic check           85.10       1873.7      1
7040      Mailed check           50.30        92.75      1
7041      Mailed check           67.85      4627.65      1
7042  Electronic check           59.00       3707.6      1

[7043 rows x 17 columns]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [8], in <cell line: 12>()
      8 print(df.fillna(0))
     10 bosque = RandomForestClassifier()
---> 12 bosque.fit(df[["tenure","TotalCharges"]].values, df["Churn"].values)

File ~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:327, in BaseForest.fit(self, X, y, sample_weight)
    325 if issparse(y):
    326     raise ValueError("sparse multilabel-indicator for y is not supported.")
--> 327 X, y = self._validate_data(
    328     X, y, multi_output=True, accept_sparse="csc", dtype=DTYPE
    329 )
    330 if sample_weight is not None:
    331     sample_weight = _check_sample_weight(sample_weight, X)

File ~\anaconda3\lib\site-packages\sklearn\base.py:581, in BaseEstimator._validate_data(self, X, y, reset, validate_separately, **check_params)
    579         y = check_array(y, **check_y_params)
    580     else:
--> 581         X, y = check_X_y(X, y, **check_params)
    582     out = X, y
    584 if not no_val_X and check_params.get("ensure_2d", True):

File ~\anaconda3\lib\site-packages\sklearn\utils\validation.py:964, in check_X_y(X, y, accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric, estimator)
    961 if y is None:
    962     raise ValueError("y cannot be None")
--> 964 X = check_array(
    965     X,
    966     accept_sparse=accept_sparse,
    967     accept_large_sparse=accept_large_sparse,
    968     dtype=dtype,
    969     order=order,
    970     copy=copy,
    971     force_all_finite=force_all_finite,
    972     ensure_2d=ensure_2d,
    973     allow_nd=allow_nd,
    974     ensure_min_samples=ensure_min_samples,
    975     ensure_min_features=ensure_min_features,
    976     estimator=estimator,
    977 )
    979 y = _check_y(y, multi_output=multi_output, y_numeric=y_numeric)
    981 check_consistent_length(X, y)

File ~\anaconda3\lib\site-packages\sklearn\utils\validation.py:746, in check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, estimator)
    744         array = array.astype(dtype, casting="unsafe", copy=False)
    745     else:
--> 746         array = np.asarray(array, order=order, dtype=dtype)
    747 except ComplexWarning as complex_warning:
    748     raise ValueError(
    749         "Complex data not supported\n{}\n".format(array)
    750     ) from complex_warning

ValueError: could not convert string to float: ''
```

In [15]:

```
bosque = RandomForestClassifier()

bosque.fit(df[["tenure","MonthlyCharges"]].values, df["Churn"].values)

print(bosque.score(df[["tenure","MonthlyCharges"]].values,df["Churn"].values))

print(cross_val_score(bosque,df[["tenure","MonthlyCharges"]].values,df["Churn"].values,
                      cv=5).mean())
```

```
0.9598182592645179
0.5571464288018582
```