Model Building

[52]:

**import**

**pandas**

**as**

**pd**

**from**

**sklearn**

**import**

metrics

**from**

**sklearn**

**.**

**model\_selection**

**import**

train\_test\_split

**from**

**sklearn**

**.**

**metrics**

**import**

recall\_score

**from**

**sklearn**

**.**

**metrics**

**import**

classification\_report

**from**

**sklearn**

**.**

**metrics**

**import**

confusion\_matrix

**from**

**sklearn**

**.**

**tree**

**import**

DecisionTreeClassifier

[53]:

df

=

pd

.

read\_csv(

'

tel\_churn.csv

'

)

[54]:

df

.

head()

[54]: Unnamed: 0 SeniorCitizen MonthlyCharges TotalCharges Churn \

1. 0 0 29.85 29.85 0
2. 1 0 56.95 1889.50 0
3. 2 0 53.85 108.15 1
4. 3 0 42.30 1840.75 0 4 4 0 70.70 151.65 1

gender\_Female gender\_Male Partner\_No Partner\_Yes Dependents\_No … \

1. True False False True True …
2. False True True False True …
3. False True True False True …
4. False True True False True …
5. True False True False True …

PaymentMethod\_Bank transfer (automatic) \

1. False
2. False
3. False
4. True
5. False

PaymentMethod\_Credit card (automatic) PaymentMethod\_Electronic check \

|  |  |  |
| --- | --- | --- |
| 0 | False | True |
| 1 | False | False |
| 2 | False | False |
| 3 | False | False |

4 False True

PaymentMethod\_Mailed check tenure\_group\_1 - 12 tenure\_group\_13 - 24 \

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | False | True | False |
| 1 | True | False | False |
| 2 | True | True | False |
| 3 | False | False | False |
| 4 | False | True | False |

tenure\_group\_25 - 36 tenure\_group\_37 - 48 tenure\_group\_49 - 60 \

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | False | False | False |
| 1 | True | False | False |
| 2 | False | False | False |
| 3 | False | True | False |
| 4 | False | False | False |

tenure\_group\_61 - 72

1. False
2. False
3. False
4. False
5. False

[5 rows x 52 columns]

[55]:

df

=

df

.

drop(

'

Unnamed: 0

'

, axis

=

1

)

[56]:

df

.

head()

[56]: SeniorCitizen MonthlyCharges TotalCharges Churn gender\_Female \

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 29.85 | 29.85 | 0 | True |
| 1 | 0 | 56.95 | 1889.50 | 0 | False |
| 2 | 0 | 53.85 | 108.15 | 1 | False |
| 3 | 0 | 42.30 | 1840.75 | 0 | False |
| 4 | 0 | 70.70 | 151.65 | 1 | True |

gender\_Male Partner\_No Partner\_Yes Dependents\_No Dependents\_Yes … \

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | False | False | True | True | False … |
| 1 | True | True | False | True | False … |
| 2 | True | True | False | True | False … |
| 3 | True | True | False | True | False … |
| 4 | False | True | False | True | False … |

PaymentMethod\_Bank transfer (automatic) \

1. False
2. False
3. False
4. True
5. False

PaymentMethod\_Credit card (automatic) PaymentMethod\_Electronic check \

|  |  |  |
| --- | --- | --- |
| 0 | False | True |
| 1 | False | False |
| 2 | False | False |
| 3 | False | False |
| 4 | False | True |

PaymentMethod\_Mailed check tenure\_group\_1 - 12 tenure\_group\_13 - 24 \

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | False | True | False |
| 1 | True | False | False |
| 2 | True | True | False |
| 3 | False | False | False |
| 4 | False | True | False |

tenure\_group\_25 - 36 tenure\_group\_37 - 48 tenure\_group\_49 - 60 \

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | False | False | False |
| 1 | True | False | False |
| 2 | False | False | False |
| 3 | False | True | False |
| 4 | False | False | False |

tenure\_group\_61 - 72

1. False
2. False
3. False
4. False
5. False

[5

rows x 51 columns

]

[57]:

*# creating x and y variable*

*# X is independent variable and Y is dependent variable which is churn*

x

=

df

.

drop(

'

Churn

'

, axis

=

1

)

x

[57]: SeniorCitizen MonthlyCharges TotalCharges gender\_Female gender\_Male \

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | |  | 0 | 29.85 | | 29.85 | |  | True | False |
| 1 | |  | 0 | 56.95 | | 1889.50 | |  | False | True |
| 2 | |  | 0 | 53.85 | | 108.15 | |  | False | True |
| 3 | |  | 0 | 42.30 | | 1840.75 | |  | False | True |
| 4 | |  | 0 | 70.70 | | 151.65 | |  | True | False |
| … | | … |  | … | | … | | … | … |  |
| 7027 | |  | 0 | 84.80 | | 1990.50 | |  | False | True |
| 7028 | 0 | | | | 103.20 | | 7362.90 | | True | False |
| 7029 | 0 | | | | 29.60 | | 346.45 | | True | False |
| 7030 | 1 | | | | 74.40 | | 306.60 | | False | True |
| 7031 | 0 | | | | 105.65 | | 6844.50 | | False | True |

Partner\_No Partner\_Yes Dependents\_No Dependents\_Yes PhoneService\_No \

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | False True True | False |  | True |
| 1 | True False True | False |  | False |
| 2 | True False True | False |  | False |
| 3 | True False True | False |  | True |
| 4 | True False True | False |  | False |
| … | … … … … |  | … |  |
| 7027 | False True False | True |  | False |
| 7028 | False True False | True |  | False |
| 7029 | False True False | True |  | True |
| 7030 | False True True | False |  | False |
| 7031 | True False True  … PaymentMethod\_Bank transfer (automatic) \ | False |  | False |
| 0 | … False |  |  |  |
| 1 | … False |  |  |  |
| 2 | … False |  |  |  |
| 3 | … True |  |  |  |
| 4 | … False |  |  |  |
| … | … … |  |  |  |
| 1. … False 2. … False 3. … False 4. … False 5. … True | |

PaymentMethod\_Credit card (automatic) PaymentMethod\_Electronic check \

|  |  |  |
| --- | --- | --- |
| 0 | False | True |
| 1 | False | False |
| 2 | False | False |
| 3 | False | False |
| 4 | False | True |
| … | … | … |
| 7027 | False | False |
| 7028 | True | False |
| 7029 | False | True |
| 7030 | False | False |
| 7031 | False | False |

PaymentMethod\_Mailed check tenure\_group\_1 - 12 tenure\_group\_13 - 24 \

1. False True False
2. True False False

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | True | True |  | False |
| 3 | False | False |  | False |
| 4 | False | True |  | False |
| … | … | … | … |  |
| 7027 | True | False |  | True |
| 7028 | False | False |  | False |
| 7029 | False | True |  | False |
| 7030 | True | True |  | False |
| 7031 | False | False |  | False |

tenure\_group\_25 - 36 tenure\_group\_37 - 48 tenure\_group\_49 - 60 \

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | False | False |  | False |
| 1 | True | False |  | False |
| 2 | False | False |  | False |
| 3 | False | True |  | False |
| 4 | False | False |  | False |
| … | … | … | … |  |
| 7027 | False | False |  | False |
| 7028 | False | False |  | False |
| 7029 | False | False |  | False |
| 7030 | False | False |  | False |
| 7031 | False | False |  | False |

tenure\_group\_61 - 72

1. False
2. False
3. False
4. False
5. False

… …

1. False
2. True 7029 False
3. False
4. True

[7032

rows x 50 columns

]

[58]:

*# creating x and y variable*

*# X is independent variable and Y is dependent variable which is churn*

y

=

df[

'

Churn

'

]

y

[58]:

0

0

1. 0
2. 1
3. 0
4. 1

..

* 1. 0
  2. 0
  3. 0
  4. 1
  5. 0

Name: Churn, Length: 7032, dtype: int64

[59]: *# Train,test split* x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y, test\_size = 0.2)

**0.0.1**

**Decision Tree Classifier**

[60]:

model\_dt

=

DecisionTreeClassifier(criterion

=

'

gini

'

, random\_state

=

100

,

␣

↪

max\_depth

=

6

, min\_samples\_leaf

=

8

)

[61]:

model\_dt

.

fit(x\_train,y\_train)

[61]: DecisionTreeClassifier(max\_depth=6, min\_samples\_leaf=8, random\_state=100)

[62]:

y\_pred

=

model\_dt

.

predict(x\_test)

[63]:

y\_pred

[63]: array([1, 0, 0, …, 0, 1, 1], dtype=int64)

[64]:

print

(

classification\_report(y\_test,y\_pred

))

precision recall f1-score support

* + 1. 0.81 0.88 0.84 1013
    2. 0.60 0.47 0.52 394

accuracy 0.76 1407

macro avg 0.70 0.67 0.68 1407

weighted avg 0.75 0.76 0.75 1407

[65]:

print

(

confusion\_matrix(y\_test, y\_pred

))

[[889 124]

[210 184]]

[ ]:

*# Accuracy 0.76261*

1.Low accuracy on the above model results, because data is highly imbalanced. To solve this need to perform up sampling or down sampling

[66]:

**from**

**imblearn**

**.**

**combine**

**import**

SMOTEENN

[67]:

sm

=

SMOTEENN()

x\_resampled,y\_resampled

=

sm

.

fit\_resample(x,y)

[68]:

*# Train,test split*

xr\_train,xr\_test,yr\_train,yr\_test

=

train\_test\_split(x\_resampled,y\_resampled,

␣

↪

test\_size

=

0.2

)

[69]:

model\_dt\_smote

=

DecisionTreeClassifier(criterion

=

'

gini

'

, random\_state

=

100

,

␣

↪

max\_depth

=

6

, min\_samples\_leaf

=

8

)

[70]:

model\_dt\_smote

.

fit(xr\_train,yr\_train)

[70]: DecisionTreeClassifier(max\_depth=6, min\_samples\_leaf=8, random\_state=100)

[71]: y\_pred\_smote = model\_dt\_smote.predict(xr\_test)

[72]: print(classification\_report(yr\_test,y\_pred\_smote, labels = [0,1]))

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall f1-score | support |
| 0 | 0.96 | 0.87 0.91 | 520 |
| 1 | 0.90 | 0.97 0.94 | 657 |
| accuracy |  | 0.93 | 1177 |
| macro avg | 0.93 | 0.92 0.92 | 1177 |
| weighted avg | 0.93 | 0.93 0.93 | 1177 |

[73]: print(confusion\_matrix(yr\_test, y\_pred\_smote))

[[452 68]

[ 19 638]]

[ ]:

*# Accuracy 0.92608*

# 0.0.2 Random forest Classifier

[75]: **from sklearn.ensemble import** RandomForestClassifier

[77]: model\_rf = RandomForestClassifier(n\_estimators=100, criterion='gini',␣

↪random\_state=100, max\_depth=6, min\_samples\_leaf=8) model\_rf.fit(x\_train,y\_train) y\_pred\_rf = model\_rf.predict(x\_test)

[79]: print(classification\_report(y\_test,y\_pred\_rf, labels = [0,1]))

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall f1-score | support |
| 0 | 0.80 | 0.92 0.85 | 1013 |
| 1 | 0.66 | 0.41 0.50 | 394 |
| accuracy |  | 0.77 | 1407 |
| macro avg | 0.73 | 0.66 0.68 | 1407 |
| weighted avg | 0.76 | 0.77 0.76 | 1407 |

[90]:

print

(

confusion\_matrix(y\_test, y\_pred\_rf

))

[[929 84]

[234 160]]

[ ]:

*# Accuracy 0.77398*

2

.Applying SMOTEENN lib

[81]:

sm

=

SMOTEENN()

x\_resampled,y\_resampled

=

sm

.

fit\_resample(x,y)

[84]:

*# Train,test split*

xr\_train,xr\_test,yr\_train,yr\_test

=

train\_test\_split(x\_resampled,y\_resampled,

␣

↪

test\_size

=

0.2

)

[85]:

model\_smote\_rf

=

RandomForestClassifier(n\_estimators

=

100

, criterion

=

'

gini

'

,

␣

↪

random\_state

=

100

, max\_depth

=

6

, min\_samples\_leaf

=

8

)

[86]:

model\_smote\_rf

.

fit(xr\_train,yr\_train)

[86]: RandomForestClassifier(max\_depth=6, min\_samples\_leaf=8, random\_state=100)

[87]: y\_pred\_smote\_rf = model\_smote\_rf.predict(xr\_test)

[88]: print(classification\_report(yr\_test,y\_pred\_smote\_rf, labels = [0,1]))

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall f1-score | support |
| 0 | 0.95 | 0.90 0.93 | 536 |
| 1 | 0.92 | 0.96 0.94 | 648 |
| accuracy |  | 0.93 | 1184 |
| macro avg | 0.94 | 0.93 0.93 | 1184 |
| weighted avg | 0.93 | 0.93 0.93 | 1184 |

[89]: print(confusion\_matrix(yr\_test, y\_pred\_smote\_rf))

[[482 54]

[ 24 624]]

[ ]:

*# Accuracy 0.93412*

# 0.0.3 Saving the model

[91]:

**import**

**pickle**

[92]:

filename

=

'

model\_rf.sav

'

[93]:

pickle

.

dump(model\_smote\_rf,

open

(

filename,

'

wb

'

))

[95]:

load\_model

=

pickle

.

load(

open

(

filename,

'

rb

'

))

[96]:

load\_model

.

score(xr\_test,yr\_test)

[96]:

0.9341216216216216