2.03. knn

October 14, 2024

1 Predictive Modeling

[17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import httpimport
import joblib

from imblearn.over_sampling import SMOTE
from pathlib import Path
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split, RandomizedSearchCV,␣
↪GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
```

[18]:
```
# Import personal library
with httpimport.github_repo("junclemente", "jcds", ref="master"):
    import jcds.metrics as jm
```

[19]:
```
# Import datasets
datasets = Path("../datasets")
train_data = "training_data.csv"
val_data = "validation_data.csv"
test_data = "testing_data.csv"
train_df = pd.read_csv(datasets / train_data)
val_df = pd.read_csv(datasets / val_data)
test_df = pd.read_csv(datasets / test_data)
display(train_df.head())
display(val_df.head())
display(test_df.head())
```

```
   Undergrad_Degree Work_Experience  Employability_Before      Status  \
0  Computer Science              No            185.174286      Placed
1       Engineering              No            206.867959  Not Placed
2               Art              No            234.881837  Not Placed
```

```
3           Finance              No            173.900408      Placed
4               Art              No            184.063980  Not Placed

   Status_enc
0           1
1           0
2           0
3           1
4           0
```

```
  Undergrad_Degree Work_Experience  Employability_Before      Status  \
0         Business             Yes            261.272959      Placed
1      Engineering              No            173.558776  Not Placed
2          Finance              No            205.074388      Placed
3         Business             Yes            230.526020      Placed
4         Business              No            229.000000  Not Placed

   Status_enc
0           1
1           0
2           1
3           1
4           0
```

```
   Undergrad_Degree Work_Experience  Employability_Before      Status  \
0           Finance              No            168.775918      Placed
1          Business             Yes            195.508673      Placed
2  Computer Science              No            260.760510      Placed
3               Art              No            231.892551  Not Placed
4  Computer Science             Yes            400.000000      Placed

   Status_enc
0           1
1           1
2           1
3           0
4           1
```

1.1 Setup Training and Validation dataframes

[20]:
```
# Variables to use for predictive modeling
variables = ["Undergrad_Degree", "Work_Experience", "Employability_Before"]
target = "Status_enc"
```

[21]:
```
# Setup train, val, and test dataframes
X_train = train_df[variables]
y_train = train_df[target]
```

```
X_val = val_df[variables]
y_val = val_df[target]

X_test = test_df[variables]
y_test = test_df[target]

# One-hot encode categorical variables
X_train = pd.get_dummies(X_train, drop_first=True)
X_val = pd.get_dummies(X_val, drop_first=True)
X_test = pd.get_dummies(X_test, drop_first=True)

# Standardize cont  / Initialize scaler
scaler = StandardScaler()
std_cols = ["Employability_Before"]
X_train[std_cols] = scaler.fit_transform(X_train[std_cols])
X_val[std_cols] = scaler.transform(X_val[std_cols])
X_test[std_cols] = scaler.transform(X_test[std_cols])

display(X_train.head())
display(X_val.head())
display(X_test.head())
```

```
   Employability_Before  Undergrad_Degree_Business  \
0             -0.800730                      False
1             -0.244034                      False
2              0.474848                      False
3             -1.090036                      False
4             -0.829222                      False

   Undergrad_Degree_Computer Science  Undergrad_Degree_Engineering  \
0                               True                         False
1                              False                          True
2                              False                         False
3                              False                         False
4                              False                         False

   Undergrad_Degree_Finance  Work_Experience_Yes
0                     False                False
1                     False                False
2                     False                False
3                      True                False
4                     False                False
```

```
   Employability_Before  Undergrad_Degree_Business  \
0              1.152088                       True
1             -1.098803                      False
2             -0.290060                      False
3              0.363071                       True
```

```
4              0.323911                        True

   Undergrad_Degree_Computer Science  Undergrad_Degree_Engineering  \
0                              False                         False
1                              False                          True
2                              False                         False
3                              False                         False
4                              False                         False

   Undergrad_Degree_Finance  Work_Experience_Yes
0                     False                True
1                     False               False
2                      True               False
3                     False                True
4                     False               False

   Employability_Before  Undergrad_Degree_Business  \
0             -1.221539                      False
1             -0.535532                       True
2              1.138938                      False
3              0.398138                      False
4              4.712054                      False

   Undergrad_Degree_Computer Science  Undergrad_Degree_Engineering  \
0                              False                         False
1                              False                         False
2                               True                         False
3                              False                         False
4                               True                         False

   Undergrad_Degree_Finance  Work_Experience_Yes
0                      True               False
1                     False                True
2                     False               False
3                     False               False
4                     False                True
```

1.1.1 Check class balance

[22]:
```
y_train.value_counts()
```

[22]:
```
Status_enc
1    348
0    228
Name: count, dtype: int64
```

[23]:
```
# Use SMOTE to balance classes
smote = SMOTE(random_state=42)
```

```
X_train, y_train = smote.fit_resample(X_train, y_train)

y_train.value_counts()
```

[23]:
```
Status_enc
1    348
0    348
Name: count, dtype: int64
```

2 K-Nearest Neighbor

2.1 RandomSearchCV

[24]:
```
knn = KNeighborsClassifier()
param_dist = {
    "n_neighbors": [2, 4, 6, 8, 10, 12],
    "weights": ["uniform", "distance"],
    "metric": ["euclidean", "manhattan", "chebyshev"],
}

random_search = RandomizedSearchCV(
    estimator=knn,
    param_distributions=param_dist,
    n_iter=20,
    cv=5,
    scoring="accuracy",
    random_state=42,
)

random_search.fit(X_train, y_train)

print(random_search.best_params_)
```

```
{'weights': 'uniform', 'n_neighbors': 4, 'metric': 'chebyshev'}
```

2.2 GridSearchCV

[25]:
```
param_grid = {
    "n_neighbors": [
        2,
        3,
        4,
        5,
        6,
        7,
    ],
    "weights": ["uniform", "distance"],
    "metric": ["euclidean", "manhattan", "chebyshev"],
```

```
}

grid_search = GridSearchCV(
    estimator=knn, param_grid=param_grid, cv=5, scoring="accuracy"
)

grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
```

```
{'metric': 'manhattan', 'n_neighbors': 3, 'weights': 'uniform'}
```

2.3 Prediction Model

[26]:
```
knn_model = KNeighborsClassifier(
    metric="manhattan",
    n_neighbors=3,
    weights="uniform",
)
knn_model.fit(X_train, y_train)

y_pred = knn_model.predict(X_val)
```

[27]:
```
cm = confusion_matrix(y_val, y_pred)
jm.mc_confusion(cm)
```

```
Confusion Matrix:
[[153   8]
 [  6 217]]
```

[27]:

	Class 0	Class 1
Accuracy	0.96354	0.96354
Error rate	0.03646	0.03646
Sensitivity (Recall)	0.95031	0.97309
Specificity	0.97309	0.95031
Precision	0.96226	0.96444
F1	0.95625	0.96875
F2	0.95268	0.97135
F0.5	0.95985	0.96616

2.4 Test

[28]:
```
test_pred = knn_model.predict(X_test)
cm_test = confusion_matrix(y_test, test_pred)
jm.mc_confusion(cm_test)
```

```
Confusion Matrix:
[[ 92   3]
 [  5 140]]
```

[28]:

	Class 0	Class 1
Accuracy	0.96667	0.96667
Error rate	0.03333	0.03333
Sensitivity (Recall)	0.96842	0.96552
Specificity	0.96552	0.96842
Precision	0.94845	0.97902
F1	0.95833	0.97222
F2	0.96436	0.96819
F0.5	0.95238	0.97629

3 Export Model

[29]:
```
models = Path("../models")
joblib.dump(knn_model, models / "k_nearest_neighbor_model.pkl")
```

[29]:
```
['../models/k_nearest_neighbor_model.pkl']
```

[ ]:
```

```