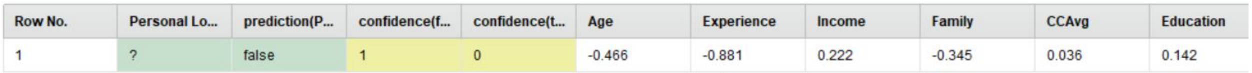
Exercise : k-NN

From Rapidminer

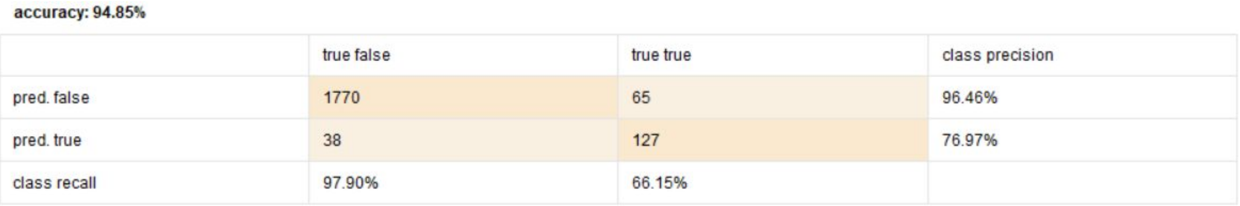
- Perform a k-NN classification with all predictors except ID and ZIP using k = 1. How would this customer be classified?



Row No.	Personal Lo...	prediction(P...	confidence(f...	confidence(t...	Age	Experience	Income	Family	CCAvg	Education
1	?	false	1	0	-0.466	-0.881	0.222	-0.345	0.036	0.142

Ans new_customer would classify in false.

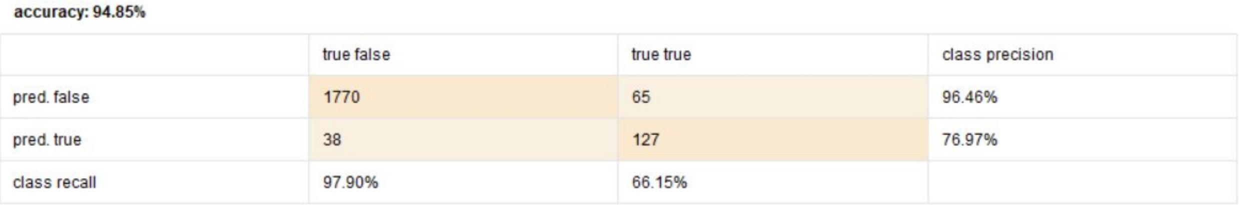
- Partition the data into training (60%) and validation (40%) sets. Show the classification matrix for the validation data that results by varying k.
    - k = 1



accuracy: 94.85%

	true false	true true	class precision
pred. false	1770	65	96.46%
pred. true	38	127	76.97%
class recall	97.90%	66.15%	

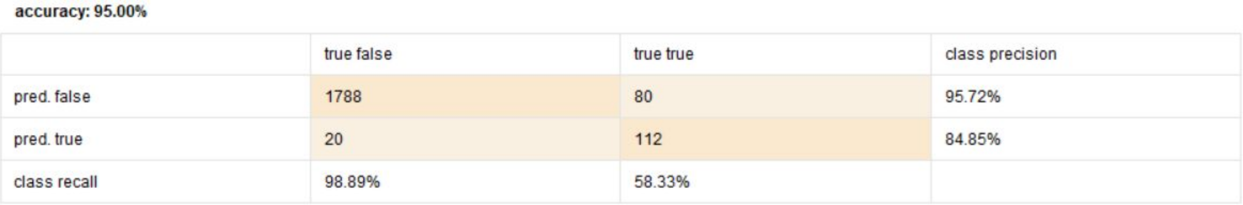
- k = 2



accuracy: 94.85%

	true false	true true	class precision
pred. false	1770	65	96.46%
pred. true	38	127	76.97%
class recall	97.90%	66.15%	

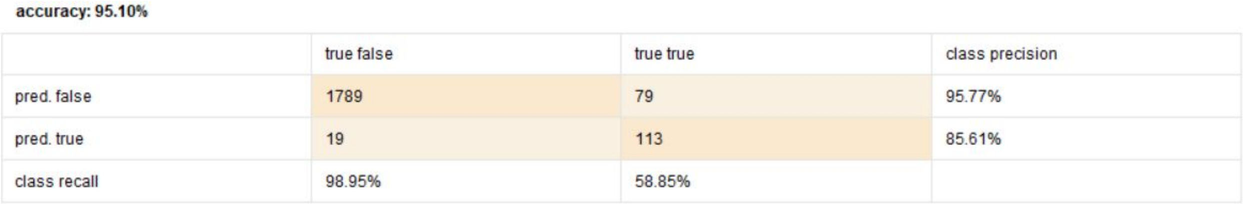
- k = 3



accuracy: 95.00%

	true false	true true	class precision
pred. false	1788	80	95.72%
pred. true	20	112	84.85%
class recall	98.89%	58.33%	

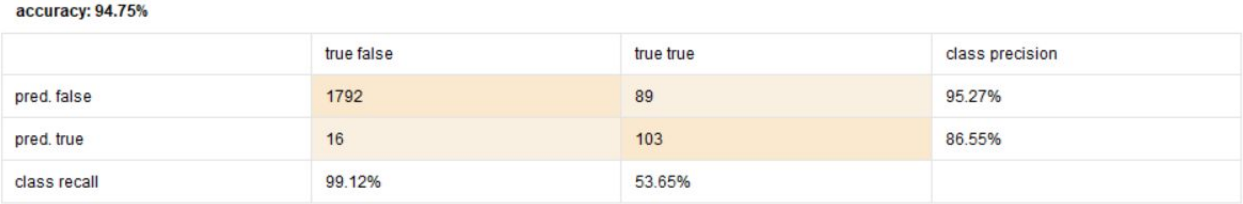
- k = 4



accuracy: 95.10%

	true false	true true	class precision
pred. false	1789	79	95.77%
pred. true	19	113	85.61%
class recall	98.95%	58.85%	

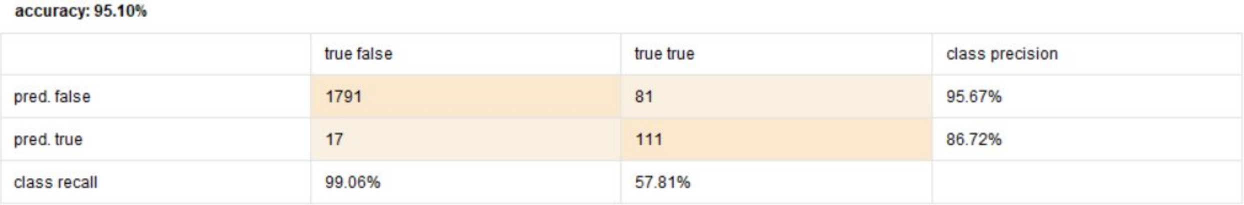
- k = 5



accuracy: 94.75%

	true false	true true	class precision
pred. false	1792	89	95.27%
pred. true	16	103	86.55%
class recall	99.12%	53.65%	

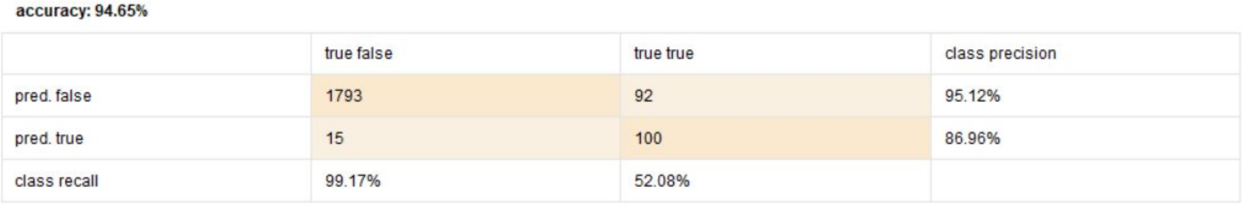
- k = 6



accuracy: 95.10%

	true false	true true	class precision
pred. false	1791	81	95.67%
pred. true	17	111	86.72%
class recall	99.06%	57.81%	

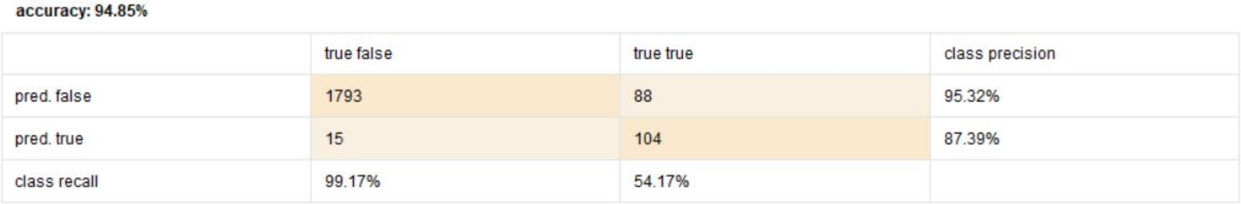
- k = 7



accuracy: 94.65%

	true false	true true	class precision
pred. false	1793	92	95.12%
pred. true	15	100	86.96%
class recall	99.17%	52.08%	

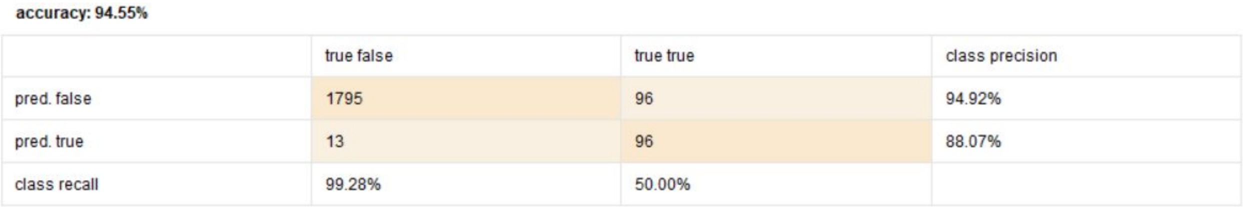
- k = 8



accuracy: 94.85%

	true false	true true	class precision
pred. false	1793	88	95.32%
pred. true	15	104	87.39%
class recall	99.17%	54.17%	

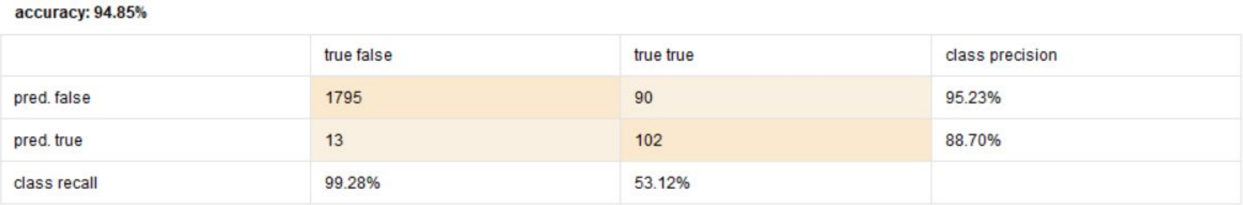
- k = 9



accuracy: 94.55%

	true false	true true	class precision
pred. false	1795	96	94.92%
pred. true	13	96	88.07%
class recall	99.28%	50.00%	

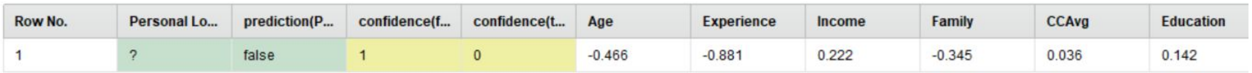
- k = 10



accuracy: 94.85%

	true false	true true	class precision
pred. false	1795	90	95.23%
pred. true	13	102	88.70%
class recall	99.28%	53.12%	

- Using the best k, how would this customer be classified?

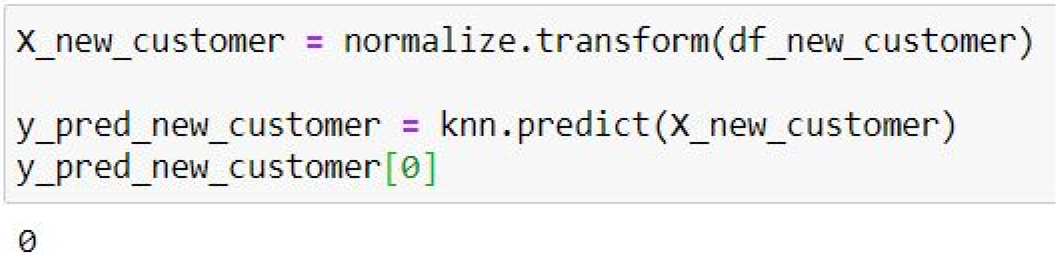


Row No.	Personal Lo...	prediction(P...	confidence(f...	confidence(t...	Age	Experience	Income	Family	CCAvg	Education
1	?	false	1	0	-0.466	-0.881	0.222	-0.345	0.036	0.142

Ans new_customer would classify in false. (k = 4)

From Python

- Perform a k-NN classification with all predictors except ID and ZIP using k = 1. How would this customer be classified?

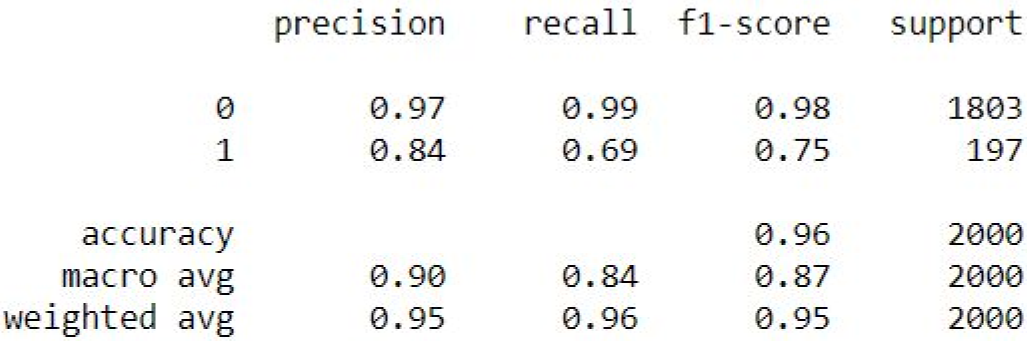


```
X_new_customer = normalize.transform(df_new_customer)

y_pred_new_customer = knn.predict(X_new_customer)
y_pred_new_customer[0]
```

```
0
```

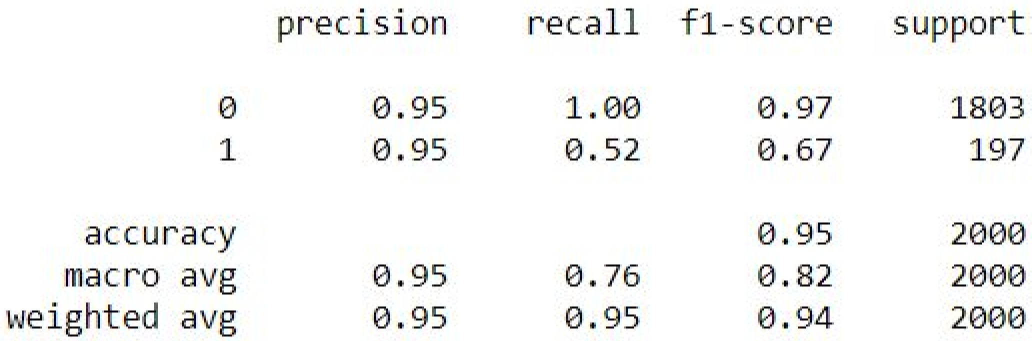
Ans new_customer would classify in false.

- Partition the data into training (60%) and validation (40%) sets. Show the classification matrix for the validation data that results by varying k.
    - k = 1



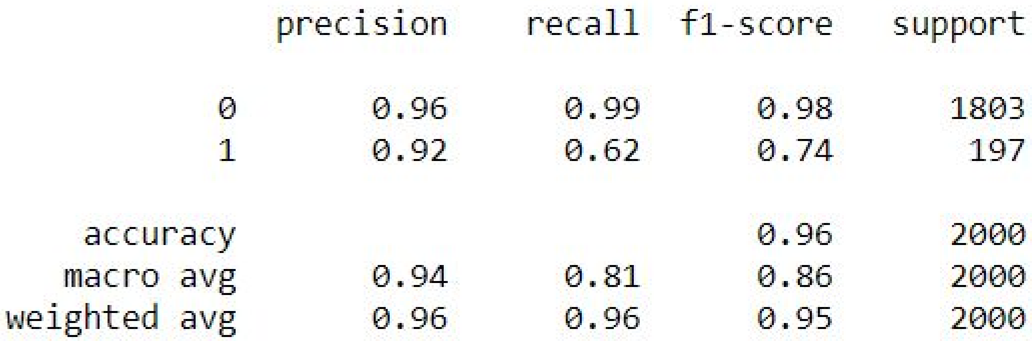
```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1803
           1       0.84      0.69      0.75       197

    accuracy                           0.96      2000
   macro avg       0.90      0.84      0.87      2000
weighted avg       0.95      0.96      0.95      2000
```

- k = 2



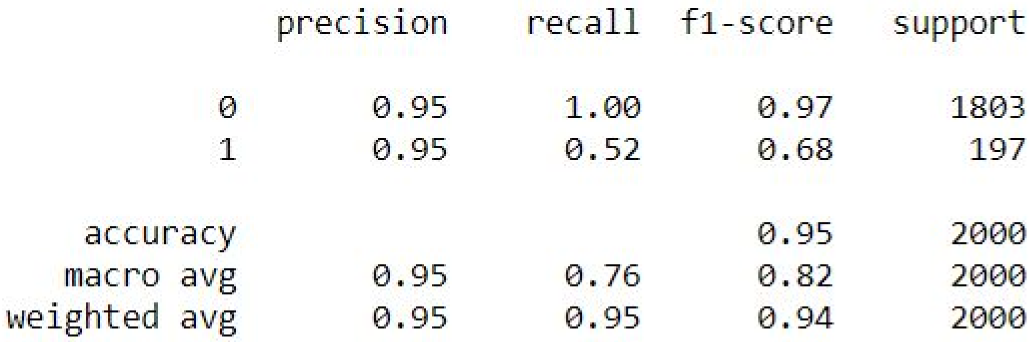
```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      1803
           1       0.95      0.52      0.67       197

    accuracy                           0.95      2000
   macro avg       0.95      0.76      0.82      2000
weighted avg       0.95      0.95      0.94      2000
```

- k = 3



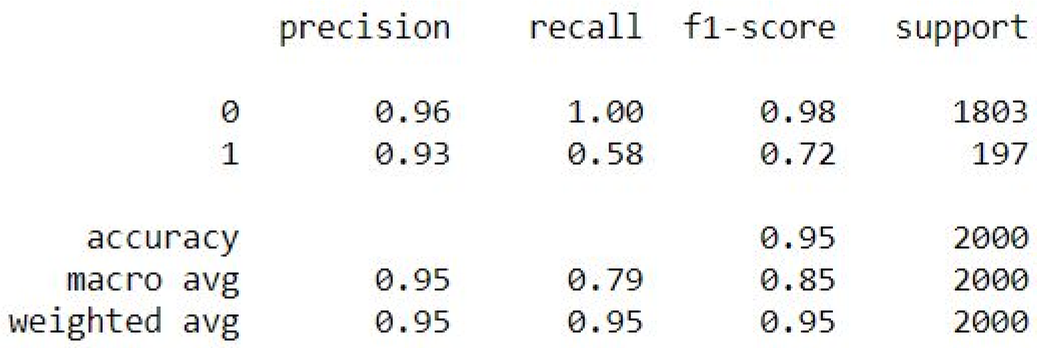
```
              precision    recall  f1-score   support

           0       0.96      0.99      0.98      1803
           1       0.92      0.62      0.74       197

    accuracy                           0.96      2000
   macro avg       0.94      0.81      0.86      2000
weighted avg       0.96      0.96      0.95      2000
```

- k = 4



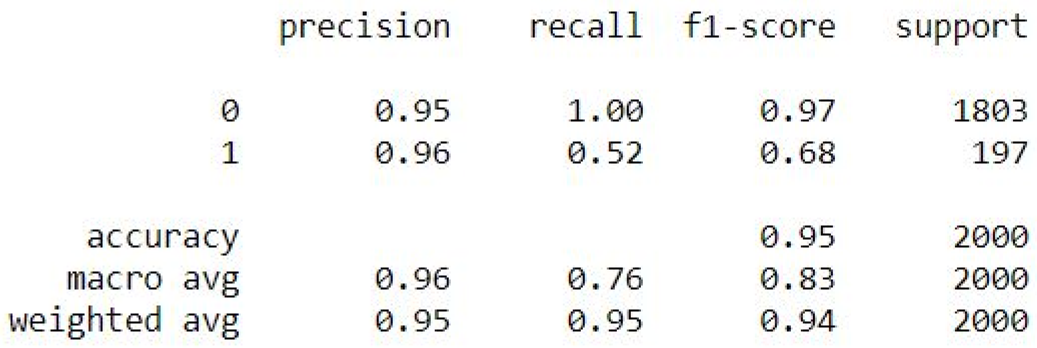
```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      1803
           1       0.95      0.52      0.68       197

    accuracy                           0.95      2000
   macro avg       0.95      0.76      0.82      2000
weighted avg       0.95      0.95      0.94      2000
```

- k = 5



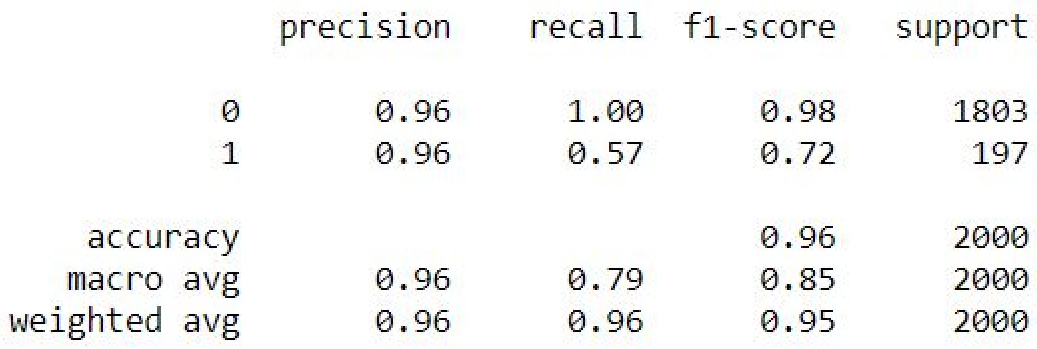
```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1803
           1       0.93      0.58      0.72       197

    accuracy                           0.95      2000
   macro avg       0.95      0.79      0.85      2000
weighted avg       0.95      0.95      0.95      2000
```

- k = 6



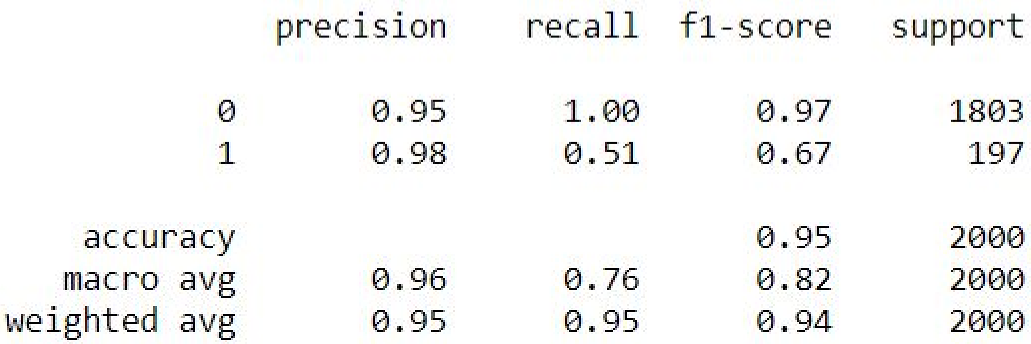
```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      1803
           1       0.96      0.52      0.68       197

    accuracy                           0.95      2000
   macro avg       0.96      0.76      0.83      2000
weighted avg       0.95      0.95      0.94      2000
```

- k = 7



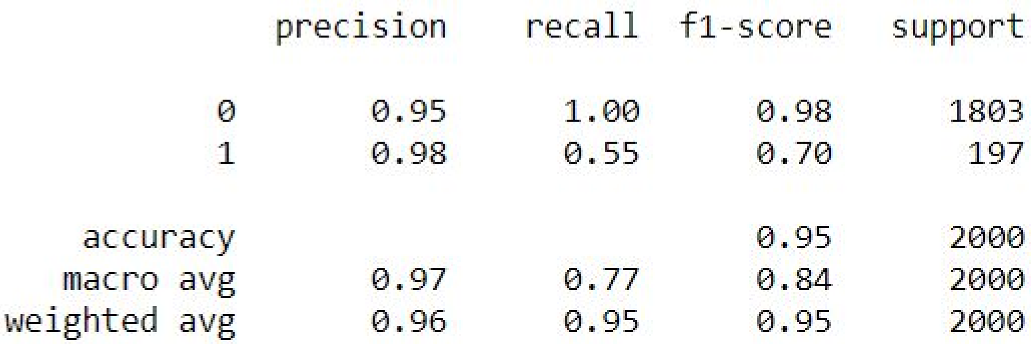
```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1803
           1       0.96      0.57      0.72       197

    accuracy                           0.96      2000
   macro avg       0.96      0.79      0.85      2000
weighted avg       0.96      0.96      0.95      2000
```

- k = 8



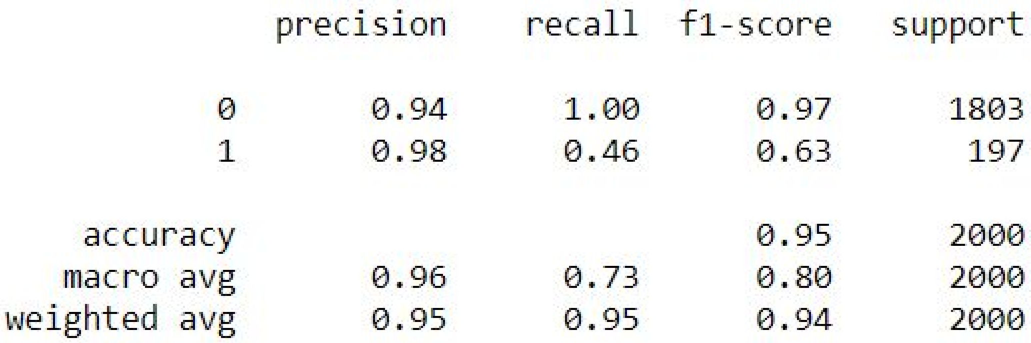
```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      1803
           1       0.98      0.51      0.67       197

    accuracy                           0.95      2000
   macro avg       0.96      0.76      0.82      2000
weighted avg       0.95      0.95      0.94      2000
```

- k = 9



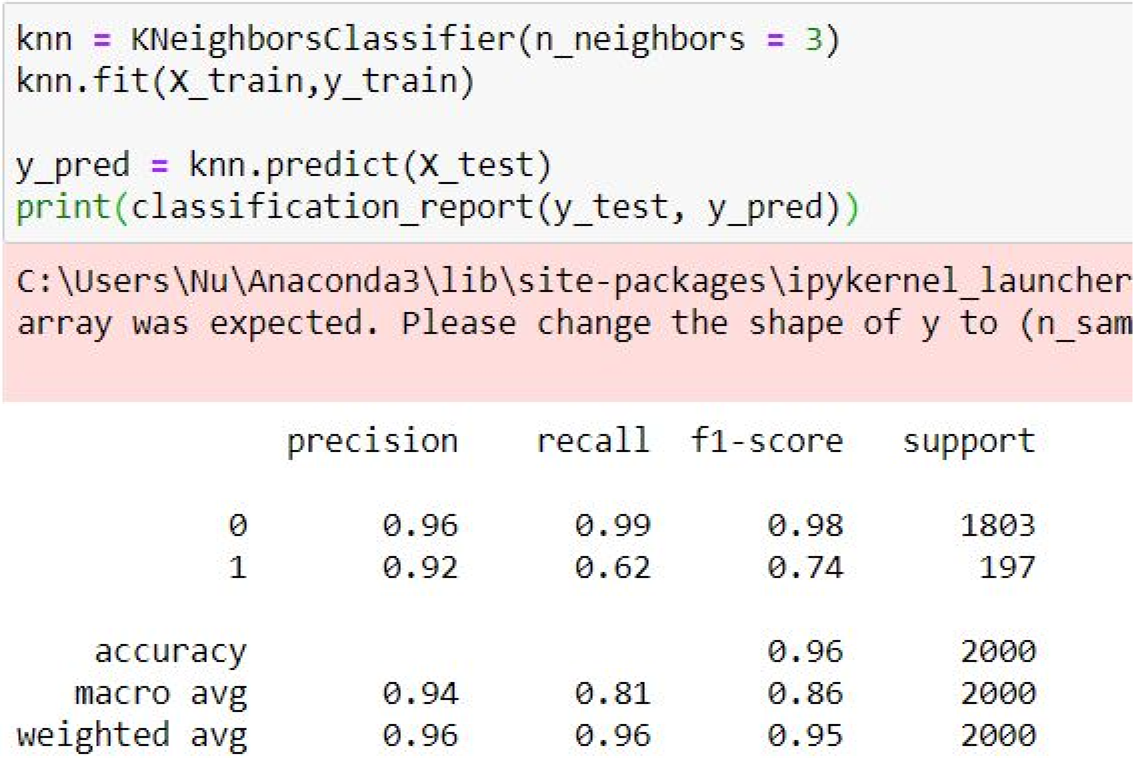
```
              precision    recall  f1-score   support

           0       0.95      1.00      0.98      1803
           1       0.98      0.55      0.70       197

    accuracy                           0.95      2000
   macro avg       0.97      0.77      0.84      2000
weighted avg       0.96      0.95      0.95      2000
```

- k = 10



```
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1803
           1       0.98      0.46      0.63       197

    accuracy                           0.95      2000
   macro avg       0.96      0.73      0.80      2000
weighted avg       0.95      0.95      0.94      2000
```

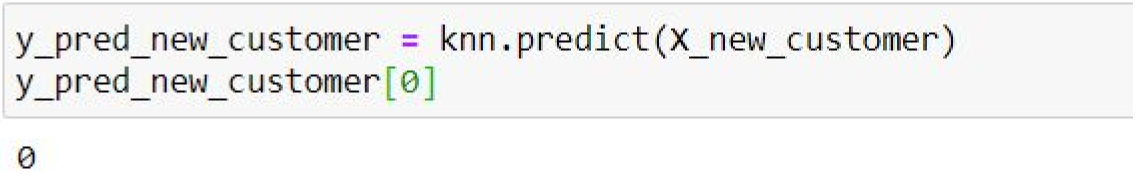
- Using the best k, how would this customer be classified?



```
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train,y_train)

y_pred = knn.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
C:\Users\Nu\Anaconda3\lib\site-packages\ipykernel_launcher
array was expected. Please change the shape of y to (n_sam
```

```
              precision    recall  f1-score   support

           0       0.96      0.99      0.98      1803
           1       0.92      0.62      0.74       197

    accuracy                           0.96      2000
   macro avg       0.94      0.81      0.86      2000
weighted avg       0.96      0.96      0.95      2000
```



```
y_pred_new_customer = knn.predict(X_new_customer)
y_pred_new_customer[0]
```

```
0
```

Ans new_customer would classify in false. (k = 3)