==========

PCA/Feature Selection using Python's sklearn library

Name: Yunjung Ham (Yunee)

I. Feature Selection and Performance Comparison

For the bankruptcy data (*'data.csv'*),[1] you are assigned to reduce the number of features for your machine learning models. Use a random forest model to obtain a subset of features by calculating the feature importances. Use **30 percent of the data for testing**. Use **0.015 for the parameter *threshold*** of *SelectFromModel.* Compare the performances of the two Random Forest classification models - one using all the features and the other using a subset of the features.

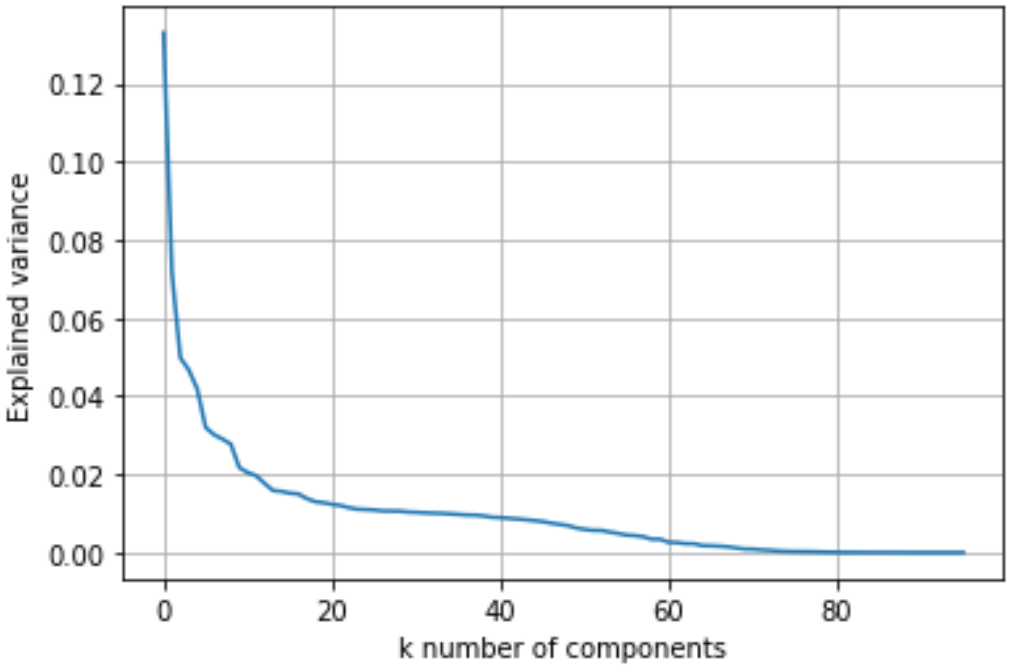
---

1. Shape of the original feature data.

```
In [19]: df.shape
Out[19]: (6819, 96)
```

---

2. Graph for PCA.

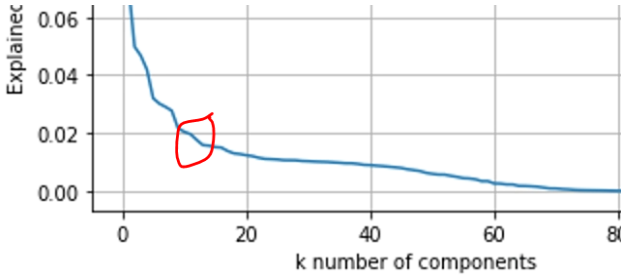


---

[1] This dataset is available at https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction. The target data are in the first column ('Bankrupt?') and the rest of the columns contain the feature data.

3. # of components

- Looking at the graph, we can see the apparent elbow forming around 16 and 17 components. I have examined the explained variance and lecture video and decided to go with 17. We can see diminishing returns as we increase beyond 17 components in our explained variance ratio as well.




```
In [10]: ev
Out[10]:
array([1.33042166e-01, 7.14465593e-02, 4.98367820e-02, 4.67479217e-02,
       4.19014322e-02, 3.20225577e-02, 3.01736399e-02, 2.90447308e-02,
       2.77447020e-02, 2.17370993e-02, 2.04660876e-02, 1.96530569e-02,
       1.77015032e-02, 1.58731924e-02, 1.56520395e-02, 1.51904806e-02,
       1.50085780e-02, 1.38279479e-02, 1.30104538e-02, 1.27572697e-02,
       1.23498325e-02, 1.20564755e-02, 1.14881093e-02, 1.10985600e-02,
       1.10009084e-02, 1.08883190e-02, 1.06603272e-02, 1.06275640e-02,
       1.06004458e-02, 1.03706253e-02, 1.02927816e-02, 1.01646543e-02,
       1.00868539e-02, 1.00290756e-02, 9.88952674e-03, 9.76477906e-03,
```

- Diminishing returns after 17 components

---

4. Show two classification reports (generated by *sklearn.metrics.classification_report*) by the random forest classification algorithm - one with the original data and the other with the components from your PCA analaysis.

Classification Report - original				
	Precision	Recall	F1-score	Support
0	0.97	1.00	0.99	1320
1	0.86	0.14	0.24	44
Accuracy			0.97	1364
Macro avg	0.97	0.57	0.61	1364
weighted avg	0.97	0.97	0.96	1364

Classification Report - pca				
	Precision	Recall	F1-score	Support
0	0.97	1.00	0.98	1320
1	0.67	0.09	0.16	44
Accuracy			0.97	1364
Macro avg	0.82	0.54	0.57	1364
weighted avg	0.96	0.97	0.96	1364

5. **Findings about the PCA for analyzing this dataset.**

- Looking at our reports, we can notice that accuracy for our original dataset and reduced dataset are both at 97%, which is a high rate. Smaller dataset with 17 selected features were still able to maintain the high accuracy rate compared to our original dataset with much more features.

Precision and recall rates for predicting our non-bankruptcy predictions (0) were same for our original and smaller dataset that we tested. However, we can see that precision and recall rates are both lower in predicting bankruptcy (1).

Feature Selection

---

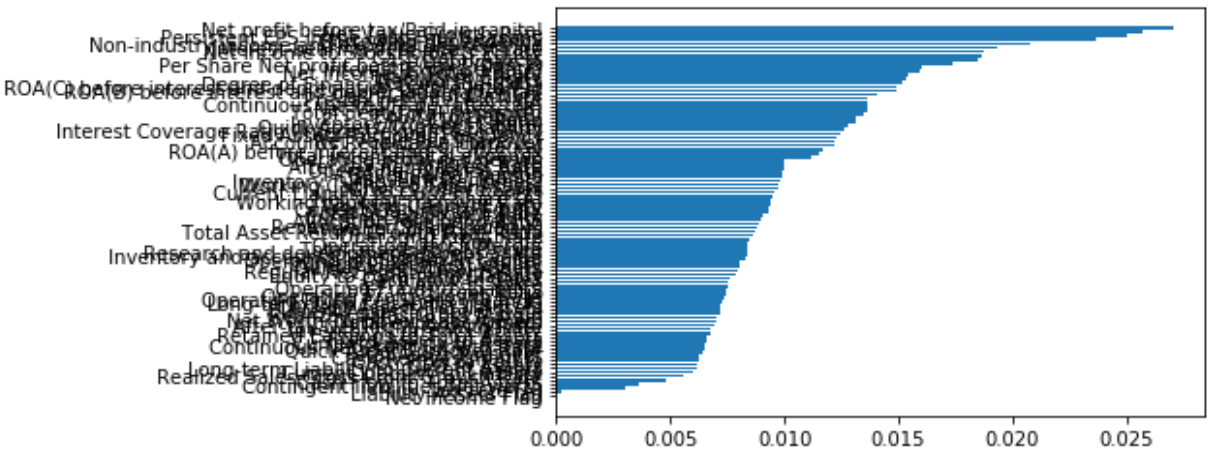
1. **Show the first five feature names.**

   In [7]: print(fn[0:5])

   Index([' ROA(C) before interest and depreciation before interest',
   ' ROA(A) before interest and % after tax',
   ' ROA(B) before interest and depreciation after tax',
   ' Operating Gross Margin',
   ' Realized Sales Gross Margin'], dtype='object')

---

2. **Horizontal bar chart between the features and the <u>sorted</u> importance.[2]**



---

3. **a.Show the number of the selected features.**

   In [22]: selector = SelectFromModel(estimator=RandomForestClassifier(),threshold=0.015)
   ...: X_reduced = selector.fit_transform(X,y)
   ...: selector.threshold_
   ...: selected_TF = selector.get_support()
   ...: print(f'\n** {selected_TF.sum()} features are selected.')

[2] Even if the feature names on the plot are not legible, do not worry about that.

**** 16 features are selected.**

---

b. Show the first five names of those selected features.

In [28]: **print(selected_features[0:5])**

[' Continuous interest rate (after tax)',
' Interest-bearing debt interest rate',
' Net Value Per Share (B)',
' Persistent EPS in the Last Four Seasons',
' Per Share Net profit before tax (Yuan ¥)']

---

5. classification reports (generated by *sklearn.metrics.classification_report*) by the two random forest classification models - one with the original data and the other with the reduced set of data.

Classification Report – Original Data				
	Precision	Recall	F1-score	Support
0	0.97	1.00	0.99	1980
1	0.85	0.17	0.28	66
Accuracy			0.97	2046
Macro avg	0.91	0.58	0.63	2046
weighted avg	0.97	0.97	0.96	2046

Classification Report – Reduced Data				
	Precision	Recall	F1-score	Support
0	0.97	1.00	0.98	1980
1	0.60	0.18	0.28	66
Accuracy			0.97	2046
Macro avg	0.79	0.59	0.63	2046
weighted avg	0.96	0.97	0.96	2046

---

6. Discuss your findings about using a reduced set of features for analyzing the bankruptcy data. Include in your discussion how you interpret *accuracy, recall, and precision* of those two models.

- Looking at the classification reports for both original and reduced data, we can notice that accuracy for our original dataset and reduced dataset are both very high at 97%, even with reduced data and features. Our original data had 95 features whereas the reduced data contains only 16 features.

  However, we can notice the lower precision and recall rates in both the original and the reduced data set. The precision rate is much lower at 60% for our reduced dataset compared to our

original data with 85% precision rate. Precision and recall rates for predicting our non-bankruptcy predictions (0) were same for our original and smaller dataset that we tested. However, we can see that precision and recall rates are both substantially lower in predicting bankruptcy (1). As mentioned above about lower precision rate, our predicted values are close to the expected/true values, but will be off by a similar amount.