1.1

Please see python program

1.2

['variance', 'skewness', 'curtosis', 'entropy']

μ(f1) σ(f1) μ(f2) σ(f2) μ(f3) σ(f3) μ(f4) σ(f4)

0 2.28 2.02 4.26 5.14 0.80 3.24 -1.15 2.13

1 -1.87 1.88 -0.99 5.40 2.15 5.26 -1.25 2.07

all 0.43 2.84 1.92 5.87 1.40 4.31 -1.19 2.10

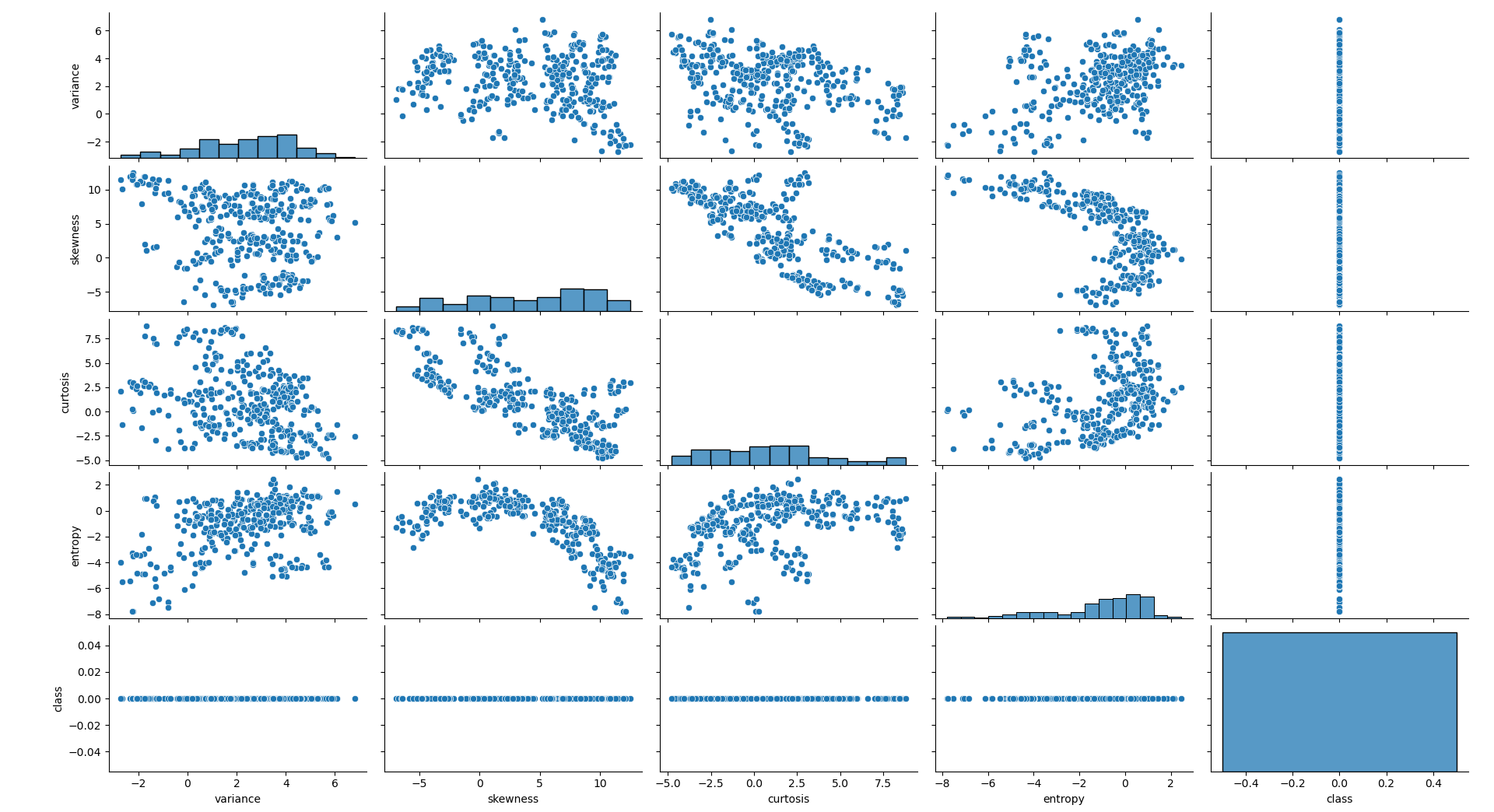
1.3

1. fake bills has a negative mean variance and skewness versus good bills has positive variance and skewness.

2. f4 has the closest mean and standard deviation when compares to other three features.

2.1

fake



good

Chart, scatter chart

Description automatically generated

2.2

# if row['variance'] <= -2 or row['skewness'] <= -5 or row['curtosis'] >= 7.5:

# test\_df.loc[index, 'predicted\_label']= 1

# else:

# test\_df.loc[index, 'predicted\_label']= 0

2.3

Please see python program

2.4 & 2.5

TP:352, TN:177, FP:124, FN:33

Accuracy: 0.7711370262390671

TPR: 0.9142857142857143

TNR: 0.5880398671096345

2.6

Yes, my simple classifier gives me an accuracy of 77%

3.1

Prediction Accuracy for k=3: 1.0

Prediction Accuracy for k=5: 0.9985422740524781

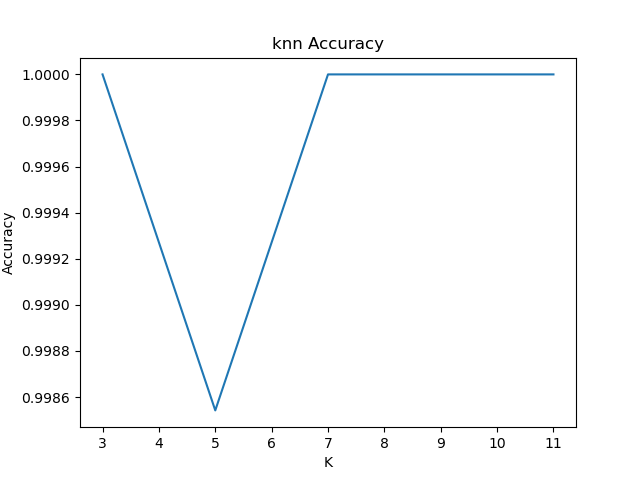
Prediction Accuracy for k=7: 1.0

Prediction Accuracy for k=9: 1.0

Prediction Accuracy for k=11: 1.0

3.2

Answer: The optimal k value for this test set is when k=3,7,9,11



3.3

TP:385, TN:301, FP:0, FN:0

Accuracy: 1.0

TPR: 1.0

TNR: 1.0

3.4

Yes, the k-NN classifier predicts nearly 100% vs my simple classifier at 77%

3.5

simple prediction result (legit is 0, fake is 1): 0

k-NN prediction result (legit is 0, fake is 1): 0

4.1

f1 truncated: Accuracy: 0.9591836734693877

f2 truncated: Accuracy: 0.9766763848396501

f3 truncated: Accuracy: 0.9693877551020408

f4 truncated: Accuracy: 0.9985422740524781

4.2

No, accuracy dropped slightly. Only f4 truncated accuracy maintained at the same level as before

4.3

By removing f1 feature, the accuracy lost that most at 3%

4.4

By removing f4 feature, the accuracy maintained at 99.85%

5.1 & 5.2

TP:380, TN:300, FP:1, FN:5

Accuracy: 0.9912536443148688

TPR: 0.987012987012987

TNR: 0.9966777408637874

5.3

es, the accuracy of logistic regression is 99.13% vs 77% of my simple classifier

5.4

No, the k-NN classifier returns a 99.85% of accuracy vs 99.13% from logistic regression

5.5

Both predicted legit (0)

6.1

f1 truncated: Accuracy: 0.8017492711370262

f2 truncated: Accuracy: 0.8994169096209913

f3 truncated: Accuracy: 0.8760932944606414

f4 truncated: Accuracy: 0.9912536443148688

6.2

No, accuracy dropped dramatically when compares to k-NN. Only f4 truncated accuracy maintained at the same level as before

6.3

By removing f1 feature, the accuracy lost the most at 19%

6.4

By removing f4 feature, the accuracy maintained at 99.13%

6.5

Yes, the logistic regression depends heavy on f1, f3, f2, then f4. Same as k-NN classifier, it depends heavily on f1, f3, f2, then f4 where f4 in both scenarios makes no differences. Another observation is that logistic regression has more significant differences when truncate the features. As result, the accuracy dropped dramatically when compares to the k-NN classifier