Rubina Iman Kabir
678 560 5414
Homework 3

1. (c)
   i. Time-domain features that are extracted from time series include statistical measures of the time series such as min, mean, max, median, mean, 1st quartile, 2nd quartile, or standard deviation.

   iii.

Table 1. Estimated standard deviations and 90% confidence interval of time domain features

| FEATURE | MIN | MAX | MEAN | MEDIAN | 1ST QUARTILE | 3RD QUARTILE |
|---|---|---|---|---|---|---|
| **ESTIMATE STANDARD DEVIATION (PANDAS)** | 10.995330 | 17.119402 | 10.510414 | 10.636667 | 10.564080 | 11.381644 |
| **ESTIMATE STANDARD DEVIATION (BOOTSTRAPPED)** | 10.933 | 17.022 | 10.451 | 10.576 | 10.5045 | 11.317 |
| **90% CONFIDENCE INVERVAL** | (8.614, 13.509) | (14.994,19.61) | (8.494,12.691) | (8.583,12.837) | (8.512,12.769) | (9.363,13.609) |

- The estimated standard deviation found with Python's bootstrapped and Pandas don't vary by a lot.
- Both Bootstrapped and Pandas standard deviation lie within the 90% confidence interval

   iv.

Table 2. Correlation between time-domain features

| | min | max | mean | median | 1qt | 3qt |
|---|---|---|---|---|---|---|
| **min** | 1.000000 | 0.839298 | 0.915141 | 0.908246 | 0.928904 | 0.891729 |
| **max** | 0.839298 | 1.000000 | 0.972534 | 0.970555 | 0.956432 | 0.980944 |
| **mean** | 0.915141 | 0.972534 | 1.000000 | 0.999057 | 0.996840 | 0.997027 |
| **median** | 0.908246 | 0.970555 | 0.999057 | 1.000000 | 0.996248 | 0.996081 |
| **1qt** | 0.928904 | 0.956432 | **0.996840** | 0.996248 | 1.000000 | 0.988678 |
| **3qt** | 0.891729 | 0.980944 | **0.997027** | 0.996081 | **0.988678** | 1.000000 |

Rubina Iman Kabir
678 560 5414
Homework 3

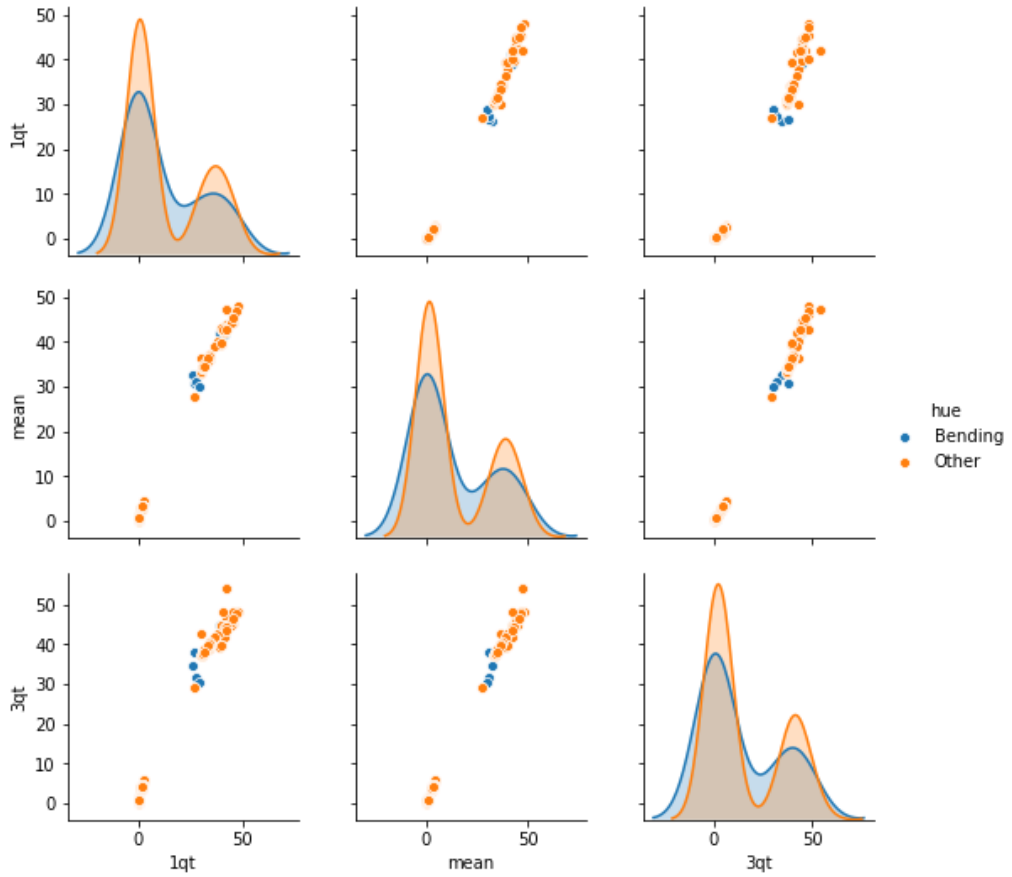
**Selected: mean, 1st quartile, and 3rd quartile**

- To select the 3 most important time-domain features I've applied a filter method where I select features with the highest correlation. I started off by finding the maximum correlation (skipping the strongest – mean and median) which is the mean and 3rd quartile. Then using the mean I selected the second highest in the column -- the 3rd quartile.

(d) Binary Classification Using Logistic Regression

i.

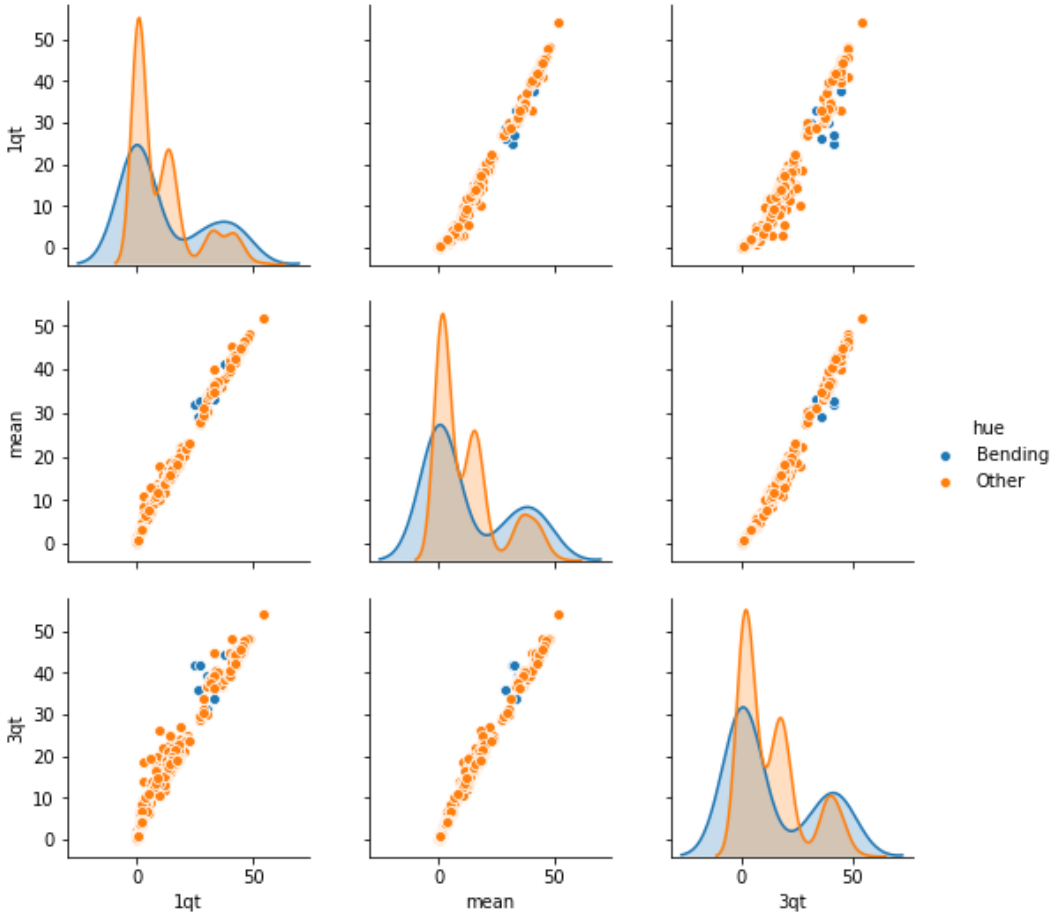
Figure1. Scatter matrix of selected time-domain features (no split).




- Using no splits we are unable to establish/visualize a clear relationship between the selected features.

ii.

Figure 2. Scatter matrix of selected-time-domain features (split = 2).




- By splitting the time-series into 2, we are able to visualize the linear relationship between the selected time-domain features compared to not splitting the data! Note how strong the relationships are.

iii.

By using sklearns RFECV to implement the feature elimination, the optimal values **(l*,p*) = (1,6).** To correctly implement cross-validation to select the model, CV was used to select the features using RFECV with 5-fold Stratified CV, where after the features were selected I cross-validated again on the found feature set using 5-fold Stratified CV. To correctly implement CV in model selection you must implement CV to find the best feature set, then implement CV on the found feature set using the left out fold and selecting the model with the best CV accuracy. The model was found using Stratified CV to take into account the class imbalance.

```
Features: ['3qt_1', 'min_5', 'max_5', 'mean_5', '1qt_5', 'max_6']
```

Rubina Iman Kabir
678 560 5414
Homework 3

iv.

Confusion matrix:

```
[[60  0]
[ 0  9]]
```

Figure 3: ROC curve for Binary Classification using Logistic Regression.

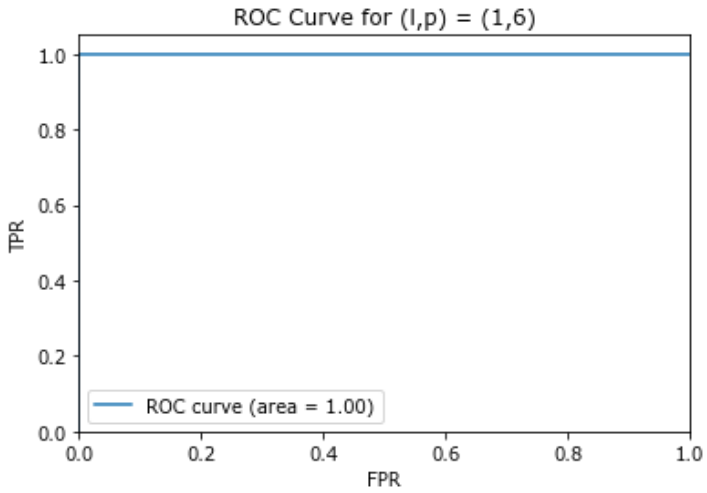



Table 3: Coefficients and p-values for (l,p) = (1,6).

| Beta | Coefficients | p-values |
|---|---|---|
| beta_1 | -0.065849 | 0.399688 |
| beta_2 | -2.234947 | 2.78982e-36 |
| beta_3 | 1.205520 | 3.22181e-12 |
| beta_4 | 1.975736 | 1.94086e-11 |
| beta_5 | 0.821936 | 3.92403e-13 |
| beta_6 | 0.949630 | 0.073004 |
| Intercept | -1.258281 | -- |

Note that when $\alpha$ = 0.05 beta_1 and beta_6 are insignificant.

v.

```
Confusion Matrix:
                [[15  0]
                [ 0  4]]
Test accuracy for l = 1 splits:  1.0
```

The model performs very well on the test set. The AUC for the test set is 1.0.

vi.

Yes, when running the dataset through statsmodels Logit we get an instability error:

```
PerfectSeparationError: Perfect separation detected, results not available
```

vii.

- Yes I do see am imbalanced class, with (class 1): (class 0) ratio of 0.13 : 0.87.
- Using case-control sampling on the training set with the features found in (d)ii, I get the following results:

```
Confusion Matrix for Sub-Sampled Training Set:
        [[12  0]
        [ 0  9]]
Train AUC with subsampling:  1.0

Confusion Matrix for Test set:
       [[15  0]
       [ 0  4]]
Test AUC with subsampling:  1.0
```

(e) Binary Classification Using *L*1-penalized logistic regression

i.

The optimal **(l*,λ*) = (3 , 1 / 21.5443)** with a CV accuracy of approximately 88%.

```
Confusion Matrix for Train Set:
        [[60  0]
        [ 0  9]]
ROC AUC For train set:  1.0

Confusion Matrix for Test Set:
       [[15  0]
       [ 0  4]]
ROC AUC For train set:  1.0
```

ii.

The L1-penalized performs faster than using recursive feature elimination and is a lot easier to implement. The AUC for both models are 1 – so performance is the same.

(f) Multi-class Classification (The Realistic Case)

i.

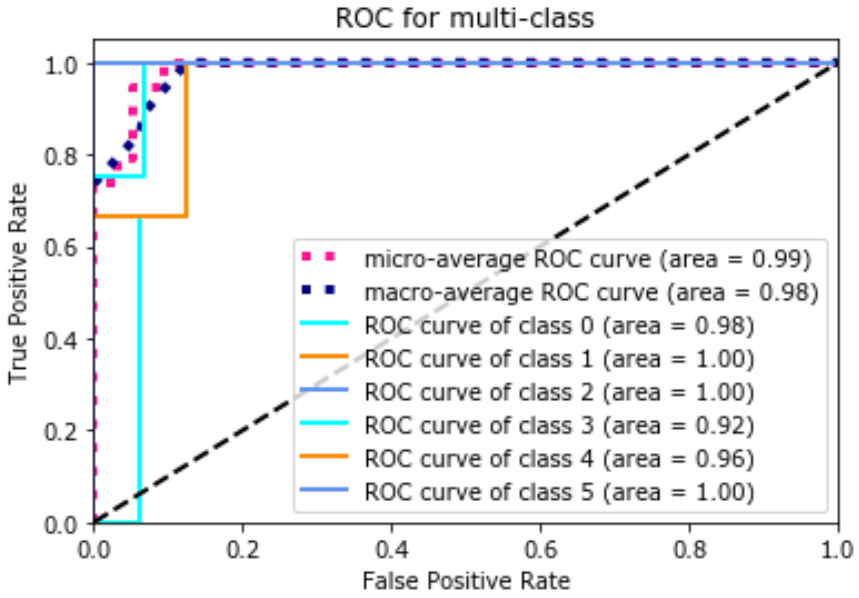
The optimal **(l*,λ*) = (4 , 1 / 2.78256)** with a CV accuracy of approximately 89%.

```
Confusion Matrix for Train Set:
       [[ 9  0  0  0  0  0]
       [ 0 12  0  0  0  0]
       [ 0  0 12  0  0  0]
       [ 0  0  0 12  0  0]
       [ 0  0  0  0 12  0]
       [ 0  0  0  0  0 12]]

Accuracy score For train set:  1.0

Confusion Matrix for Test Set:
       [[3 0 0 1 0 0]
       [0 3 0 0 0 0]
       [0 0 3 0 0 0]
       [1 0 0 0 2 0]
       [0 0 0 0 3 0]
       [0 0 0 0 0 3]]
Accuracy score For test set:  0.7894736842105263
```

Note that the performance for multinomial regression using L1 penalization on the test set is comparably less than the binary classification. Based off the ROC curves, the area under the curves for each class are high, with the lowest being for class 3 where AUC = 0.92.

Figure 4: ROC Curve for Multi-class Classification.

Rubina Iman Kabir
678 560 5414
Homework 3




ii.

Gaussian Naïve Bayes:

Best Split: 1

```
Confusion Matrix for Test Set for Gaussian:
        [[4 0 0 0 0 0]
        [0 3 0 0 0 0]
        [1 0 2 0 0 0]
        [2 0 0 1 0 0]
        [0 0 1 2 0 0]
        [0 2 0 0 0 1]]


    Accuracy on test set for Gaussian NB: 0.5789473684210527
```

Multinomial Naïve Bayes:

Best Split: 1

```
Confusion Matrix for Test Set for Multinomial:
        [[4 0 0 0 0 0]
         [0 3 0 0 0 0]
         [0 0 3 0 0 0]
         [0 0 0 2 1 0]
         [0 0 0 0 3 0]
         [0 0 0 0 0 3]]


    Accuracy on test set for Multinomial NB: 0.9473684210526315
```

iii.

Recall the accuracy scores:

Multinomial L1-Penalized Logistic Regression: `0.7894736842105263`
Gaussian Naïve Bayes Classifier: `0.5789473684210527`
Multinomial Naïve Bayes Classifier: `0.9473684210526315`

Rubina Iman Kabir
678 560 5414
Homework 3

**Multinomial Naïve Bayes Classifier** performs the best.

2. (a) We would expect the linear regression model to have a smaller RSS because the data is fitted better using degree 1 versus degree 3.

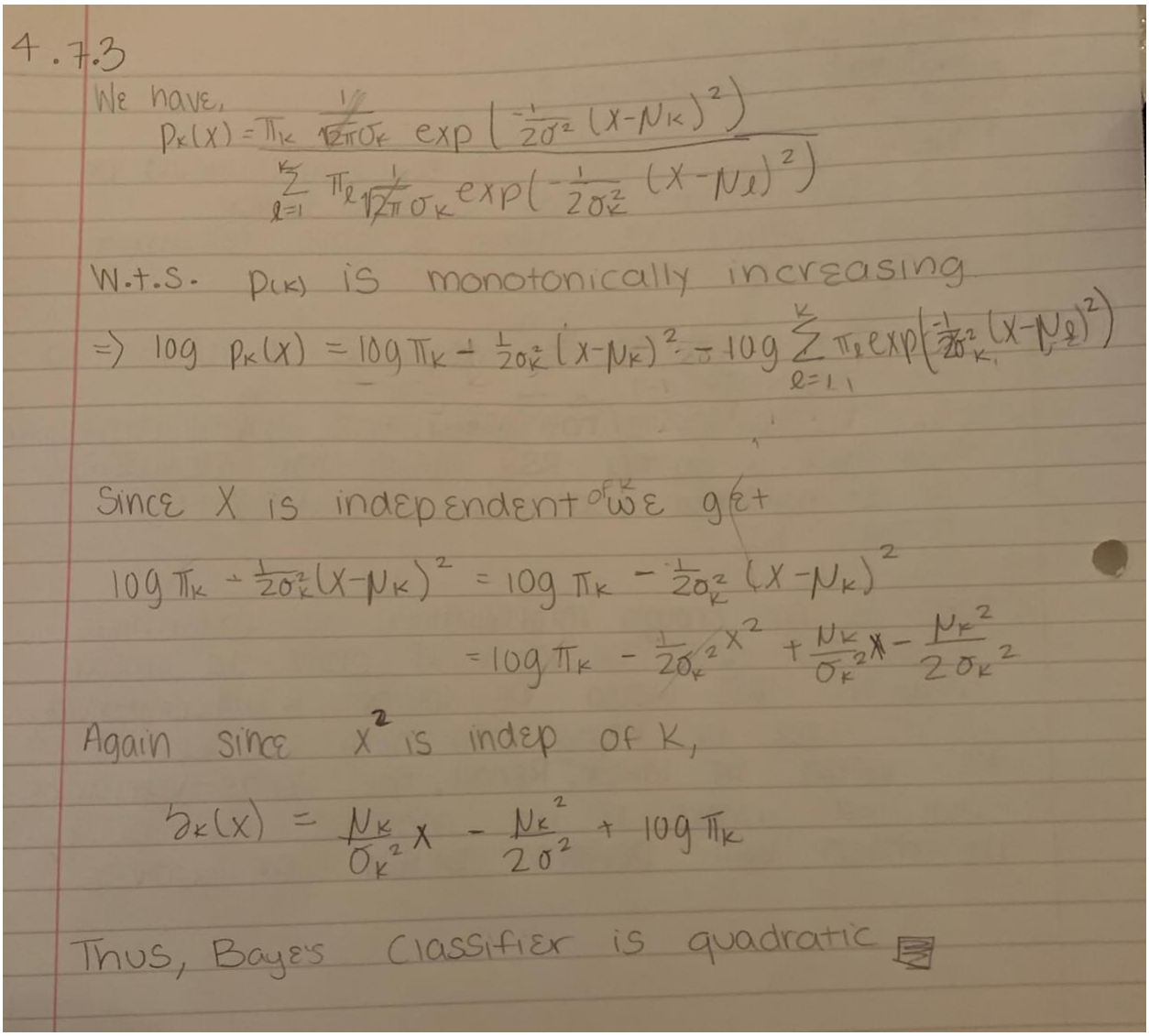
(b) The RSS for linear regression, again, would be smaller because the cubic fit would be overfitting the training data and hence resulting in a larger RSS.

(c) Since the model is not linear, the cubic regression will have a smaller RSS since, in general, as we increase flexibility on a model with a data set with anon-linear relationship between X and Y the RSS will decrease.

(d) There is not enough information to determine which RSS would be lower. For the linear case, if the true relationship is close to linear then the RSS would be lower. For the cubic case, if the true relationship is close to being non-linear then the RSS would be lower. Recall the bias-variance trade off where if we increase flexibility the model will perform better or worse.

Rubina Iman Kabir
678 560 5414
Homework 3

3.



4.7.3

We have,

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(\frac{-1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_l)^2\right)}$$

W.t.s. $p(k)$ is monotonically increasing

$$\Rightarrow \log p_k(x) = \log \pi_k - \frac{1}{2\sigma_k^2}(x-\mu_k)^2 - \log \sum_{l=1}^{K} \pi_l \exp\left(\frac{-1}{2\sigma_k^2}(x-\mu_l)^2\right)$$

Since $X$ is independent of $\varepsilon$ we get

$$\log \pi_k - \frac{1}{2\sigma_k^2}(x-\mu_k)^2 = \log \pi_k - \frac{1}{2\sigma_k^2}(x-\mu_k)^2$$

$$= \log \pi_k - \frac{1}{2\sigma_k^2}x^2 + \frac{\mu_k}{\sigma_k^2}x - \frac{\mu_k^2}{2\sigma_k^2}$$

Again since $x^2$ is indep of $k$,

$$\delta_k(x) = \frac{\mu_k}{\sigma_k^2}x - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

Thus, Bayes classifier is quadratic

Rubina Iman Kabir
678 560 5414
Homework 3

4.