CS252 S22 Final Exam

Amanda Stent

11 May 2022

Instructions

This exam consists of nine sections. Each section is worth 10 points. You may pick **any seven** of the first eight sections to answer. For extra credit, you may answer an eighth section, and/or the ninth section.

Introduction

We are going to try to replicate parts of the following paper:
Chicco, D. & Jurman, G. (2020) "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone". *BMC Medical Informatics and Decision Making*, 20:16. Retrieved from `https://doi.org/10.1186/s12911-020-1023-5`.

Where does this data come from?

The authors "analyzed a dataset containing the medical records of 299 heart failure patients collected at the Faisalabad Institute of Cardiology and at the Allied Hospital in Faisalabad (Punjab, Pakistan), during April–December 2015." (Guess what? I was born in Faisalabad, though not in this hospital!)

Are there any ethical concerns with using this data?

The dataset is described in this paper:
Ahmad, T., Munir, A., Bhatti, S.H., Aftab, M. & Raza, M.A. (2017) "Survival analysis of heart failure patients: A case study". *PLoS ONE* 12(7): e0181001. Retrieved from `https://doi.org/10.1371/journal.pone.0181001`. The authors of this paper say, "The study was approved by Institutional Review Board of Government College University, Faisalabad-Pakistan and the principles of Helsinki Declaration were followed. Informed consent was taken by the patients from whom the information on required characteristics were collected/accessed."

I retrieved this data from `https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records`. According to that site, the data is released under an Attribution 4.0 International (CC BY 4.0) license.

Here is a description of the variables in this data set:

- age: age of the patient (years)
- anaemia: if the patient has anemia (boolean)
- high blood pressure: if the patient has hypertension (boolean)
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary)
- serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean)
- time: follow-up period (days)
- death event: if the patient died during the follow-up period (boolean)

There is no personally identifiable information. Although age and sex are protected categories, we will not be attempting to predict either variable. We will ignore the time variable.

We will perform clustering, regression and classification experiments with this data. No models we train will be released; however, the authors of the study we are trying to partially replicate intend their findings to be used to help physicians make decisions about which tests to order.

Columns	[age, anaemia, creatinine_phosphokinase, diabetes, ejection_fraction, high_blood_pressure, platelets, serum_creatinine, serum_sodium, sex, smoking, DEATH_EVENT]
Shape	(299, 12)
Type	float64
Missing data	none

Table 1: Data set: Basic information

	Max	Min	Mean	Std
age	95	40	60.83	11.87
anaemia	1	0	.43	.50
creatinine_phosphokinase	7861	23	581.84	968.66
diabetes	1	0	.42	.49
ejection_fraction	80	14	38.08	11.82
high_blood_pressure	1	0	.35	.48
platelets	850000	25100	263358.03	97640.55
serum_creatinine	9.4	0.5	1.39	1.03
serum_sodium	148	113	136.63	4.41
sex	1	0	.65	.48
smoking	1	0	.32	.47
DEATH_EVENT	1	0	.32	.47

Table 2: Data set: Summary statistics

1 Look at, clean and normalize your data

1.1 (1 pt) How many data points are there in this data set? *299*

1.2 (1 pt) How many variables are there? *12*

1.3 (3 pt) Why do we need to normalize this data? *Because some of the variables have a much larger range than others (e.g. platelets vs serum_creatinine). If we don't normalize, then we won't be able to do certain types of modeling (e.g. kNN, k-means clustering) very well since they are sensitive to distance.*

1.4 (2 pt) We will z-score this data. What two basic types of transformation (e.g. translation, scaling, rotation) are involved in z-scoring? *translation by the mean, scaling by the standard deviation*

1.5 (3 pt) Write the transformation matrix for z-scoring for the three variables serum_creatinine, ejection_fraction and DEATH_EVENT. *Assuming I have first selected only those three variables from my data into a separate matrix,*

$$\begin{pmatrix} \frac{1}{std(serum\_creatinine)} & 0 & 0 & -mean(serum\_creatinine) \\ 0 & \frac{1}{std(ejection\_fraction)} & 0 & -mean(ejection\_fraction) \\ 0 & 0 & \frac{1}{std(DEATH\_EVENT)} & -mean(DEATH\_EVENT) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

or

$$\begin{pmatrix} \frac{1}{1.03} & 0 & 0 & -1.39 \\ 0 & \frac{1}{11.82} & 0 & -38.08 \\ 0 & 0 & \frac{1}{.47} & -.32 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2 Consider dimensionality reduction

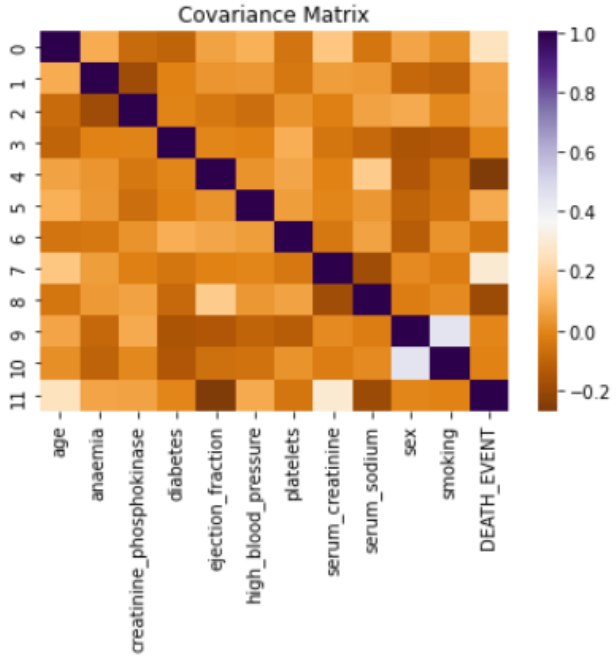


Figure 1: Covariance matrix

2.1 (2 pt) In what circumstances do we want to use dimensionality reduction? *If we have many variables, we may want to use dimensionality reduction for efficiency or to make it easier to visualize our data. We may also be able to use some types of dimensionality reduction, like PCA, to convert non-independent variables into independent variables.*

2.2 (2 pt) We use PCA for dimensionality reduction. Fill in the blanks:

(a) Before using PCA, we have to *center* the data.

(b) To fit a PCA model, we do *singular value decomposition (SVD)* on the covariance matrix computed over the data.

2.3 (2 pt) Looking at the covariance matrix in Figure 1, one pair of variables co-varies more than the rest. Which is it? *ejection_fraction and DEATH_EVENT. The color for these is darker than for the rest, indicating a covariance of around -0.2.*

2.4 (2 pt) One way to choose the number of dimensions to keep is by looking at an elbow plot. Looking at the one in Figure 2, how many dimensions should we keep for this data set in order to retain 80% of the cumulative explained variance? *I would accept 7 or 8.*

2.5 (2 pt) Considering your answer to the previous question, do you think it would be much more efficient, basically as efficient, or much less efficient, to fit classification models on the PCA-projected data? Explain your answer. *I would say basically as efficient, since we'd only go down from 12 variables to 7 (or 8), and we'd have to project the data through PCA to do that.*

3 Clustering

3.1 (2 pt) Clustering requires a distance metric. Name and define a distance metric other than Euclidean distance. *Commonly used distance metrics are:*
** Manhattan (L1 norm):* $d(\vec{a}, \vec{b}) = \sum_{i=1}^{m} |a_i - b_i|$
** Chebyshev distance (L^∞ norm):* $d(\vec{a}, \vec{b}) = max_i|a_i - b_i|$
** Minkowski distance (L^p norm):* $d(\vec{a}) = [\sum_{i=1}^{m} |a_i|^p]^{1/p}$
However, there are a lot more than that. I would accept any reasonable answer.

3.2 (2 pt) For k-means clustering, we minimize inertia. Define inertia. *inertia* $= 1/N \sum_{j=1}^{N} d(\vec{x_j}, \vec{mx_j})^2$, *where* $\vec{m_{x_j}}$ *is the centroid of the cluster that* x_j *is currently assigned to, and* d *is your chosen distance metric.*

3.3 (2 pt) k-means clustering is sensitive to the structure of the input data. In what way? How can we fix this issue? *If one variable has a range a lot larger or smaller than the others, it will dominate with many distance metrics. We fix this issue with normalization. One student said that with high dimensional data the distances are bigger, and this we would fix with PCA.*

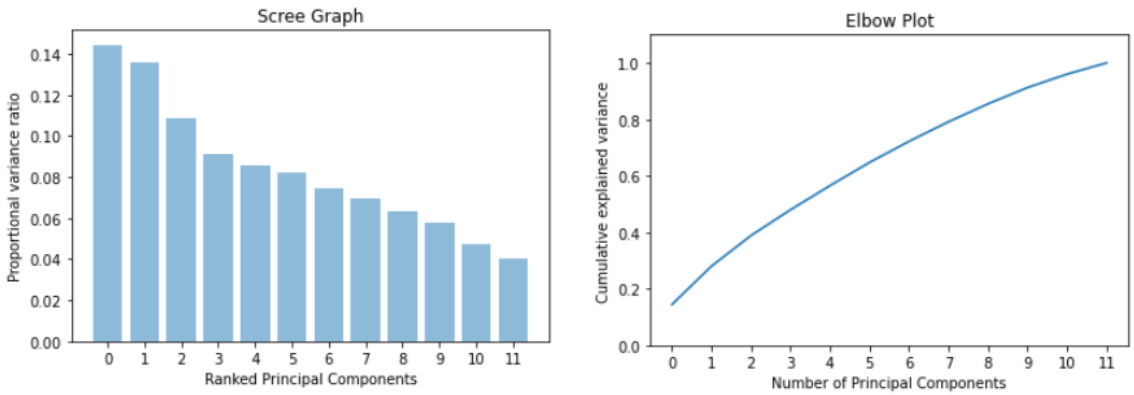


Figure 2: Scree and elbow plots for PCA

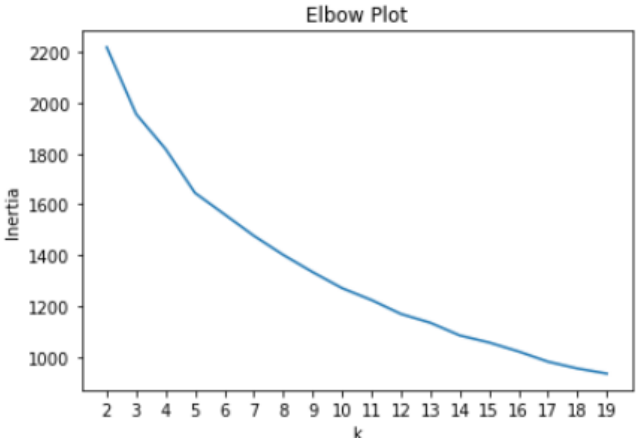


Figure 3: Elbow plot for picking k for k-means clustering

3.4 (2 pt) One way to choose k is by inspecting an elbow plot. Looking at the one in Figure 3, what would be a good value for k for this data? Why? *I would choose 3 or 5, both of which correspond to changes in the slope of the elbow plot and both of which are fairly small, so we'll be able to visualize easily.*

3.5 (2 pt) Look at the two scatter plots in Figure 4. The first plots serum_creatinine against ejection_fraction (using the z-scored data). The second plots the first principal component against the second (using the z-scored data projected through PCA). Both use colors to show DEATH_EVENT and shapes to show cluster membership. Which is clearer to you? Why? *To me, the second is clearer because (a) the data points are spread out more, instead of on top of one another like in the first plot, and (b) I can see the clusters make sense (more or less), as in data points with different shapes are in different locations in the plot.*

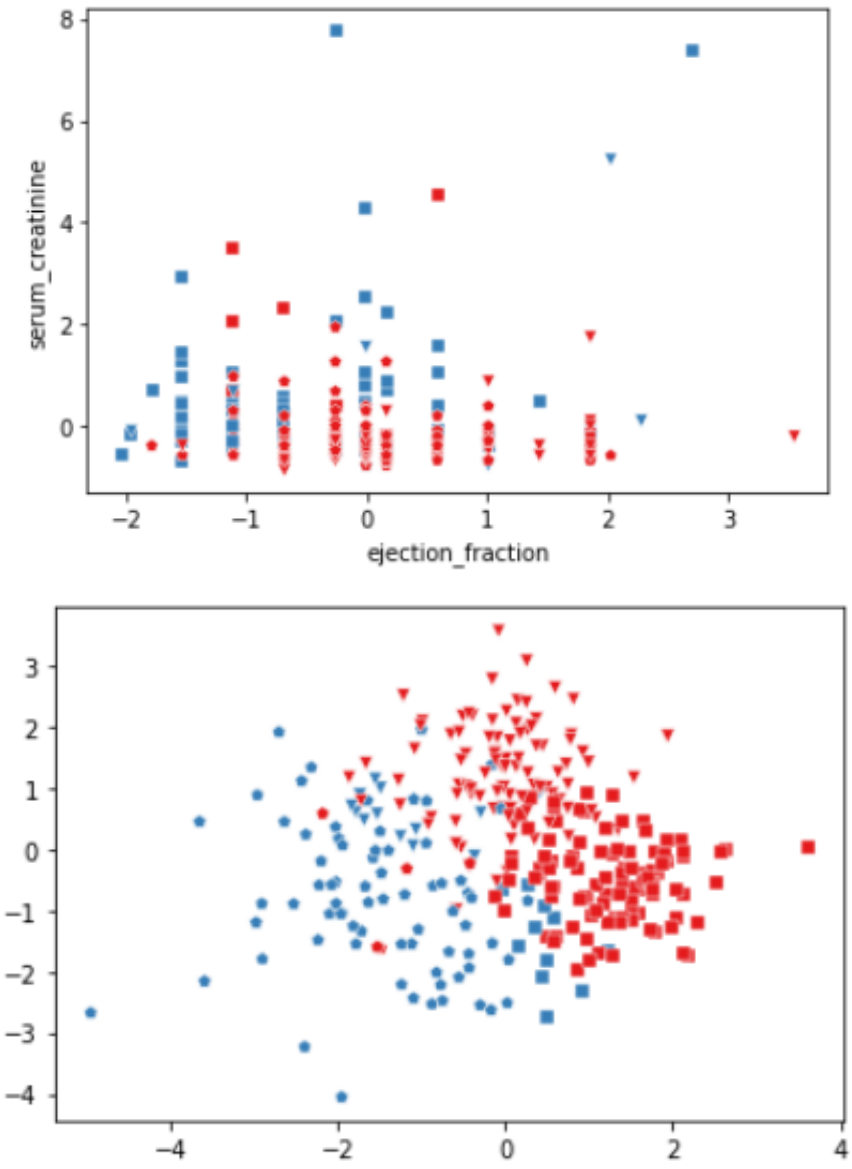


Figure 4: k-means clustering on the z-scored data (top) and the PCA-projected data (bottom)

4 Regression

4.1 (2 pt) Name and define the loss function for regression. *The loss function is MSSE (mean sum squared error):*

$$MSSE = 1/N \sum_{i=1}^{N} (r_i)^2 = 1/N \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

.

4.2 (3 pt) Least squares is one method for fitting a linear regression model. Least squares for linear regression can be summarized as solving the system of linear equations described by $\vec{y} = A \cdot \vec{c}$. In this equation:

(a) What is $\vec{y}$? *It contains the values of the dependent variable for each data point.*

(b) What do we have to do to the matrix of feature-values to construct A? *Add a leading column of 1s for the intercept.*

(c) What does $\vec{c}$ contain? *The values for the intercept and weights/slopes (one per variable) that define the line or plane fit to the data.*

4.3 (1 pt) I can also use least squares to fit a polynomial regression model. Assuming A contains only one independent variable, What do I have to do to A to fit a polynomial of degree 2? *Add a column with the square of each value for the independent variable.*

Predicted	Actual	Correct?
0.204	0.0	y
0.098	0.0	y
0.433	0.0	y
0.397	0.0	y
0.313	0.0	y
0.25	0.0	y
0.343	0.0	y
0.187	0.0	y
0.541	0.0	n
0.284	0.0	y
0.381	0.0	y
0.07	0.0	y
0.165	0.0	y
0.164	0.0	y
0.719	0.0	n
0.302	1.0	n
0.495	1.0	n
0.767	1.0	y
0.371	1.0	n
0.917	1.0	y

Table 3: Linear regression: predicted vs actual values for DEATH_EVENT

4.4 (2 pt) I evaluate the goodness of fit of a regression model by reporting R^2. R^2 for a linear regression on a held-out development subset of the heart attack data is 0.289. What does this value for R^2 mean, in terms of model performance? *It means the regression is not a very good fit to the development data, because R^2 for a good fit would be closer to 1.*

4.5 (2 pt) I can make the regression model "classify" by rounding (the absolute value of) the output. Table 3 shows predicted vs actual y values for part of the dev set from the linear regression model that has R^2 of 0.289. As a *classifier* (by rounding the predicted values) how accurate is this model? *15/20 = 75%, which is pretty good! However, if I had a trivial model that just output the majority class (0), I'd still get an accuracy of 75% on this development data, so maybe not so good!*

5 Model evaluation

In addition to accuracy, we often create confusion matrices for a classifier, and use those confusion matrices to report additional insightful measures.

5.1 (2 pt) Draw a confusion matrix. Fill in the cells with values corresponding to the data in Table 3. Label the cells corresponding to true positives, true negatives, false positives and false negatives.

DEATH_EVENT	*predicted 0*	*predicted 1*
actual 0	*13*	*2*
actual 1	*3*	*2*

. The true positives in this table are in cell (1, 1); the true negatives in cell (0, 0); the false positives in cell (0, 1) and the false negatives in cell (1, 0). You could also draw it with 1 first, then 0, in which case the locations would be different.

5.2 (2 pt) Looking at your confusion matrix, what can we say about the classes in this model? *They are unbalanced (more 0s than 1s). Also, the model does a better job of predicting the negative class (0) than the positive class (1).*

5.3 (2 pt) Define true positive rate. *TPR = TP/(TP+FN)*

5.4 (2 pt) Define false positive rate. *FPR = FP/(FP+TN)*

5.5 (2 pt) For a multiclass classifier, what is a variant on this vanilla confusion matrix that we can use? *We can make a set of one-versus-rest confusion matrices: for classes A, B, C, we would have one for A (postive) vs BC (negative), one for B (postive) vs AC (negative) and one for C (postive) vs AB (negative). Or we can make a set of one-versus-other confusion matrices: for classes A, B, C, we would have one for A vs B, one for A vs C, and one for B vs C.*

6 K-nearest neighbors

6.1 (2 pt) How does the *fit* function work for k-nearest neighbors? *You just store the training data!*

6.2 (6 pt) How does the *predict* function work? *You need a distance metric. For each data point the class of which you want to predict, calculate the distance between that data point and every data point in the training data. Sort the distances from smallest to largest and take the data points corresponding to the k smallest distances. Figure out the majority class of those k closest data points. If there is no majority class (there's a tie), you can choose at random.*

6.3 (2 pt) One way to choose a value of k is by looking at an elbow plot. Looking at the elbow plot in Figure 5, what value of k would you choose for this data and why? *6, because it is the smallest value of k that has the highest accuracy.*

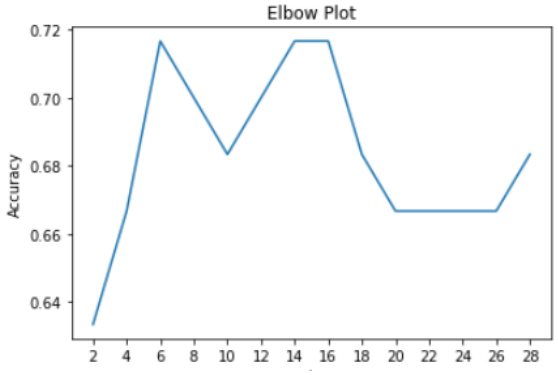


Figure 5: Elbow plot for selecting k for k-nearest neighbors

7 Naive Bayes

7.1 (2 pt) State Bayes rule. Label the parts corresponding to the posterior, prior, likelihood and evidence. $P(Y|X) = \frac{P(Y)*P(X|Y)}{P(X)}$.
* $P(Y|X)$ *is the *posterior**
* $P(Y)$ *is the *prior*; we approximate this using* $\frac{|Y|}{\sum_Y |Y|}$
* $P(X|Y)$ *is the *likelihood*; we approximate this using* $\frac{|X\&Y|}{|Y|}$
* $P(X)$ *is the *evidence* (or *normalization*); we don't need to calculate or approximate this in Naive Bayes since in Naive Bayes we are comparing the posteriors for multiple outcomes and regardless of outcome the denominator is the same.*

7.2 (2 pt) Why do we call a Naive Bayes model "naive"? What does this allow us to do? *It's "naive" because we assume each independent variable is conditionally independent of each other one. This allows us to use the chain rule to figure out likelihoods where there are multiple independent variables.*

7.3 A simple Naive Bayes model is based on relative frequencies of values of the variables in the training data.

(a) (2 pt) How can we account for values of variables we may not see for a particular class at train time? *We can use smoothing. A simple form of smoothing is Laplace smoothing: for* $x_i \in X$*:*
* *instead of* $P(Y|x_i) \sim \frac{|x_i\&Y|}{|Y|}$,
* *we use* $P(Y|x_i) \sim \frac{|x_i\&Y|+1}{|Y|+|x_i|}$.

(b) (2 pt) The estimated probabilities output via this method, for any non-trivial number of variable values, will be very small. How can we handle this? *We can calculate log priors and log likelihoods instead of raw priors and likelihoods.*

7.4 (2 pt) Why would we *not* want to fit a vanilla Naive Bayes to our data, but rather fit a Gaussian Naive Bayes? *Several of the variables in our data are quantitative continuous or quantitative discrete, so they have an infinite number of possible values. This means we can't draw a finite likelihoods table.*

8 RBF networks

8.1 (2 pt) What is a radial basis function? *"a real-valued function whose value depends only on the distance between the input and some fixed point" (per Wikipedia).*

8.2 (2 pt) In this course, what type of activation function did we define for the hidden nodes? Write it out. *We use a Gaussian,* $exp\left(-\frac{||\vec{d}-\vec{\mu_j}||^2}{2\sigma_j^2+\epsilon}\right)$*, where* $\vec{d}$ *is the data point,* $\vec{\mu_j}$ *is the prototype,* σ_j *is the hidden unit's standard deviation,* ϵ *is a small constant and* $||.||^2$ *is the squared Euclidean distance. We calculate* σ_j *as the average distance from any training data point in the* j*th cluster to the centroid of the* j*th cluster.*

8.3 (2 pt) For what types of modeling can we use a RBF network? *classification, regression*

8.4 Let's think about training a RBF network on the full set of variables in this data set to determine whether a "death event" is likely to occur.

(a) (2 pt) How many nodes will be in the input layer? *11, one for each independent variable*

(b) (2 pt) How many nodes will be in the output layer? *2, one for each value of the dependent variable*

9 The replication (EXTRA CREDIT ONLY)

The researchers in the paper compared many different modeling methods, including linear regression, k-nearest neighbors, Naive Bayes and an "artificial neural network".

- As in the paper, we compare models trained on all the variables to models trained using only the variables serum_creatinine, ejection_fraction and DEATH_EVENT.
- As in the paper, we report results averaged over 100 runs of each modeling approach.
- As in the paper, in each run we split the data into 60% train, 20% dev and 20% test. Unlike in the paper, (a) we perform stratified sampling, ensuring each class is represented in train, dev and test in proportion to its frequency in the whole data set; and (b) for modeling approaches that do not require hyperparameter tuning, we still only train on train and test on test.

.

Figure 6 displays *boxplots* across accuracy, TPR and TNR for each of the four approaches and for each combination of variables. A boxplot (or box and whisker plot) shows summary statistics for an array of numbers. In these boxplots:

- The middle horizontal line denotes the median of the array.
- The box is drawn from quartile 1 to quartile 3.
- The top line corresponds to a distance of 1.5 times the "inter quartile range" beyond quartile 3.
- The bottom line corresponds to a distance of 1.5 the "inter quartile range" beyond quartile 1.
- The dots correspond to outliers.

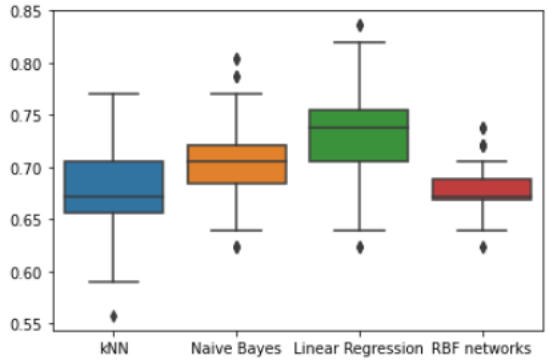
So, the top left corner boxplot shows the median, Q1, Q3, range and outliers for accuracy, across 100 runs, for models trained on the full set of variables. Below each figure is a table showing the min, max, mean and standard deviation for the metric across all 100 runs.

9.1 (2 pt) Which *two* of our approaches require hyperparameter tuning in each round? *kNN, RBF networks (for the k-means clustering).*

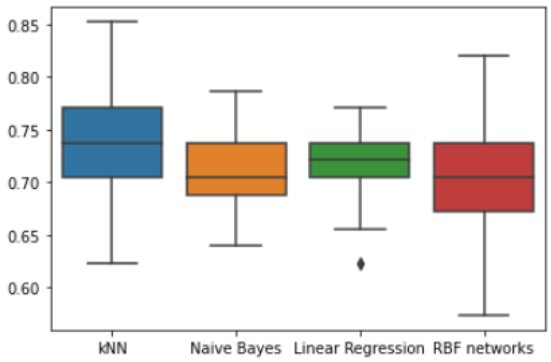
9.2 (3 pt) How would you characterize the accuracy of these modeling approaches on the *full* variable set versus on the *reduced* variable set (for example, does one variable set give clearly more accurate results)? Put another way, would you say the title of the paper is accurate? *I would say the title of the paper is accurate. Comparing the top two box plots, there's no modeling approach that is outside the whiskers of any other. This was not asked, but the TPR and TNR also don't indicate that the full feature set is better than the reduced one. One student also observed that quartiles are smaller (the boxes smaller) with the full data set, so maybe it is better.*

9.3 (3 pt) How would you characterize the differences in accuracy between the different modeling approaches (for example, is one clearly more accurate than the others)? *Comparing the models in each box plot of the top two box plots, there's no modeling approach that is outside the whiskers of any other. So no approach is clearly more accurate.*

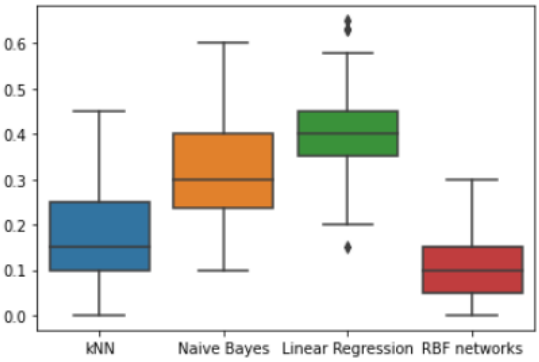
9.4 (2 pt) Comparing accuracy to the true positive rate and true negative rate, what can you say about the distribution of classes in this data set? *It's unbalanced. There are more of the negative class (0) than the positive class (1), and this leads to a higher TNR than TPR. To fix that situation (if we wanted to do better on predicting 1), we'd need to use a modeling approach that gives us probabilities of some kind, and then pick a threshold that preferred 1 (e.g. for regression-as-classification, we might pick a threshold of .4 or .3 for saying the class was "1" vs just rounding, which is a threshold of .5).*



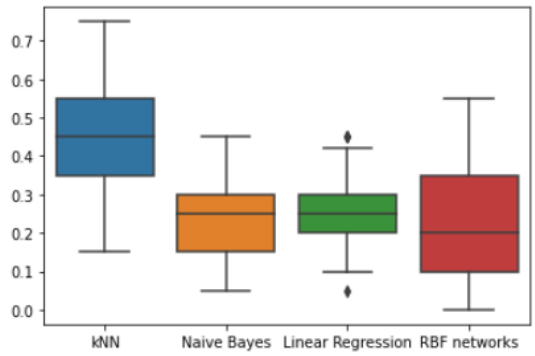
	kNN	NB	LR	RBF
max	.77	.80	.84	.74
min	.56	.62	.62	.62
mean	.68	.71	.73	.68
std	.04	.04	.04	.03



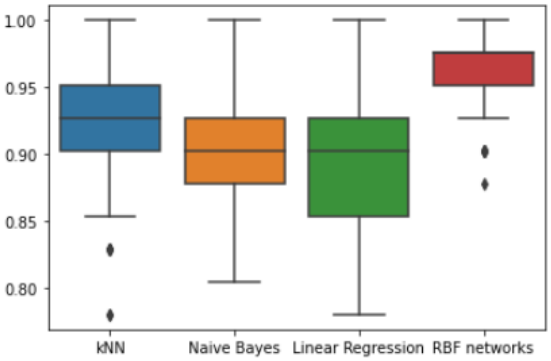
	kNN	NB	LR	RBF
max	.85	.79	.77	.82
min	.62	.64	.62	.57
mean	.74	.71	.72	.70
std	.05	.03	.03	.04



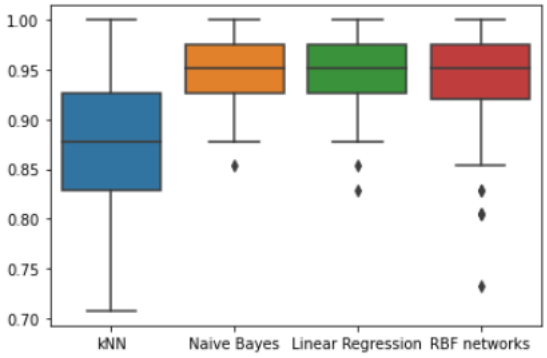
	kNN	NB	LR	RBF
max	.45	.60	.65	.30
min	.00	.10	.15	.00
mean	.17	.31	.40	.10
std	.10	.10	.10	.06



	kNN	NB	LR	RBF
max	.75	.45	.45	.55
min	.15	.05	.05	.00
mean	.45	.22	.26	.23
std	.13	.09	.07	.14



	kNN	NB	LR	RBF
max	1.00	1.00	1.00	1.00
min	.78	.80	.78	.88
mean	.92	.90	.89	.96
std	.05	.05	.05	.03



	kNN	NB	LR	RBF
max	1.00	1.00	1.00	1.00
min	.71	.85	.83	.73
mean	.88	.95	.95	.94
std	.06	.03	.04	.05

Figure 6: Accuracy (top row), true positive rate (middle row) and true negative rate (bottom row) for models trained on the full set of variables (left) and the reduced set of variables (right)