Selecting Features in a Dry Bean Dataset

Report for HarvardX PH125.9x: Data Science: Capstone

Samuel Bates

June 23, 2021

1 Introduction

In this report, I explore a few different ways to select a subset of features for training a classification model. Using a dataset of measurements performed on seven types of dry beans(Koklu and Ozkan 2020), I investigate the feature distributions in a couple of ways, and create an algorithm for selecting features that maximizes accuracy with as few features as possible. I train five models on the various sets of features, indicating the accuracy of each model on each feature set. Two of the models report the importance of features during training, so I compare them to the feature sets I construct. Lastly, I check the accuracy of the trained models on a reserved testing set.

2 Data Exploration

The dataset contains 13,611 observations. Each observation was constructed from a high-resolution photograph of a dry bean, and has 17 features. Twelve of the features are geometric, four are unknown "shape factors" $SF1 - SF4$, and one ($Class$) indicates the type of bean. The twelve geometric factors are:

1. Area (denoted by A): The number of pixels in the bean image.
2. Perimeter (P): The length of the bean boundary measured in pixels.
3. Major axis length (X[1]): The length of the longest line that can be drawn within the bean boundary.
4. Minor axis length (x): The length of the longest line perpendicular to the major axis that can be drawn within the bean boundary.
5. Aspect ratio (K): Defined as X/x.
6. Eccentricity (Ec): The eccentricity of an ellipse that has the same moments of inertia as the bean image.
7. Convex area (C): The number of pixels within the smallest convext polygon that contains the bean image.
8. Equivalent diameter (Ed): The diameter of a circle having the same area as the bean image. Expressed mathematically, $A = \pi(Ed/2)^2$, so Ed is defined as $\sqrt{4A/\pi}$.
9. Extent (Ex): The ratio of the pixels in the bounding box to the bean area.

[1]The document accompanying the dataset uses L and l for the major and minor axis lengths. I changed them to X and x, respectively, for greater readability.

10. Solidity (S): Also known as convexity. The ratio of the pixels in the convex shell to the pixels in the bean image.
11. Roundness (R): Defined as $4\pi A/P^2$.
12. Compactness (CO): Defined as Ed/X.

There are seven types of bean represented: Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira. Figure 1 shows the distribution of the features for each type of bean. I will use F_{all} to denote the full set of 16 features.

2.1 Feature selection

2.1.1 Using the definitions

The first step in feature selection relies on the feature definitions themselves: Five of the geometric features are defined in terms of other features. Four of these are explicit: aspect ratio, equivalent diameter, roundness, and compactness. In addition, solidity S is defined in terms of the "convex shell," which is similar to the "smallest convex polygon" used in the definition of C. Experimentally, I found that S is equal to A/C. These five relationships are verified to 9 decimal places by the following code:

```
near_zero <- 10^-10
all(
  max(abs(beans$K - beans$X / beans$x)) < near_zero,
  max(abs(beans$A - pi * beans$Ed^2 / 4)) < near_zero,
  max(abs(beans$S - beans$A / beans$C)) < near_zero,
  max(abs(beans$R - 4 * pi * beans$A / beans$P^2)) < near_zero,
  max(abs(beans$CO - beans$Ed / beans$X)) < near_zero)
```

```
## [1] TRUE
```

Consequently, an observation is completely defined by just 11 features: the 7 remaining geometric features plus the 4 shape factors.

2.1.2 Using correlation

Correlation allows me to eliminate another geometric feature. Figure 1 shows that the area (A) and convex area (C) distributions for each type of bean are very similar. Table 1 shows that they are very highly correlated, so C can be safely removed from the list of features. The table provides additional evidence that I may safely remove CO and Ed, as $SF3$ and P, respectively, are highly correlated with them. I will use F_{min} to denote the resulting set of 10 features $\{A, P, X, x, Ec, Ex, SF1, SF2, SF3, SF4\}$.

Table 1: The most highly-correlated features.

Feature 1	Feature 2	Correlation
A	C	0.99994
CO	SF3	0.99869
P	Ed	0.99138

Principal component analysis suggests that F_{min} may still be larger than necessary. I applied `prcomp` to the set of features and discovered that six components can explain 100% of the variability. In fact, the first two components can explain the variablity to 6 decimal places; adding four more components reduces the standard deviation to less than 1 (Table 2).

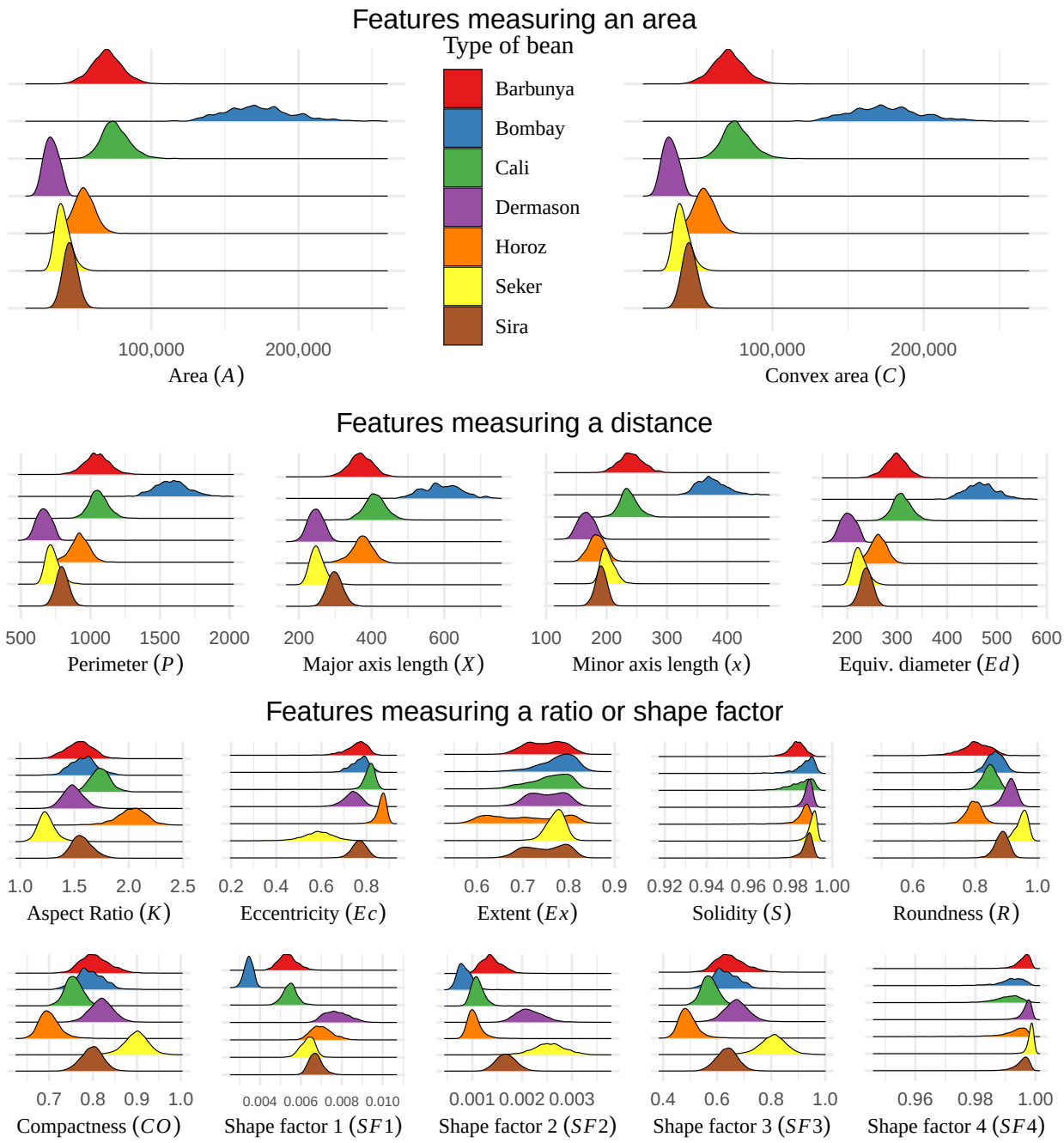

Figure 1: Distributions of the features, grouped by the type of feature.

Table 2: The first six components of principal component analysis.

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	41790.54577	231.24444	57.924	20.018	6.568	0.78991
Proportion of Variance	0.99997	0.00003	0.000	0.000	0.000	0.00000
Cumulative Proportion	0.99997	1.00000	1.000	1.000	1.000	1.00000

3 Models and Analysis

I tested five models on the training set using both the full and minimal sets of features:

- linear discriminant analysis (`lda`);
- quadratic discriminant analysis (`qda`);
- k nearest neighbors (`knn`);
- recursive partitioning (`rpart2`); and
- bootstrap aggregated trees (`treebag`).

The result appears in Figure 2. In three of the models (`lda`, `qda`, and `treebag`), the minimal set F_{min} produced a lower accuracy than the full set F_{all}. The other two models (`knn` and `rpart2`) showed F_{min} producing as high an accuracy or higher than that produced by F_{all}.

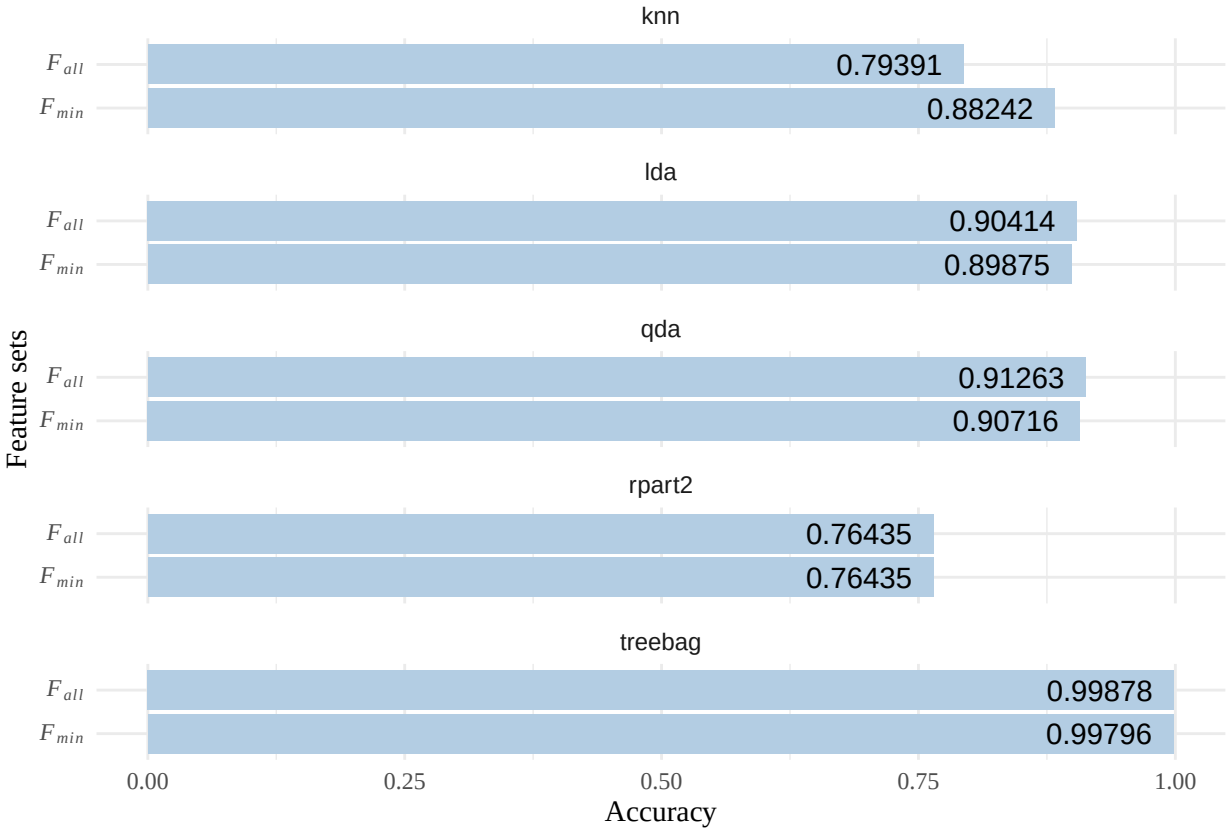

Figure 2: Accuracy of each model and feature set on the training set.

3.1 Feature selection by repeated predictions

To further explore what features are truly necessary, I wrote an algorithm that uses repeated predictions to find a minimal set of features that maximizes accuracy. The algorithm proceeds as follows:

1. Using a feature set F, calculate the accuracy of a model on each single feature f in F. Choose a feature with the highest accuracy (there may be more than one). Say it is feature f_1, and that the accuracy is A_1. I denote the set of used features with U; in this case, it is $\{f_1\}$. I denote the accuracy of the model on U with A_U.

2. For each feature g that is in $F - U$, calculate the accuracy A_g of the model on the set of features $U \cup \{g\}$. (After Step 1, this is the set $\{f_1, g\}$.)

    1. If all the values in $\{A_g | g \in F - U\}$ are less than or equal to A_U, then stop.
    2. If there is a value $A_g > A_U$, pick a feature with the largest value (again, there may be more than one) and add it to the set U. Reset A_U to be the new higher value A_g.

3. Repeat Step 2 until either there are no features left (i.e., $U = F$), or no feature produces a higher accuracy than A_U. The final set U is the desired minimal set of features.

The algorithm will clearly produce different minimal sets for different models. Less obviously, it will produce different minimal sets for different values of F. Figure 3 shows the results of running the algorithm on each model with $F = F_{all}$ or $F = F_{min}$. The accuracies for the full set and minimal set are shown for comparison.

3.2 Feature selection by model-dependent importance

One last exploration involves the idea of variable importance. When the caret package `train` function is executed on the last two models, `rpart2` and `treebag`, the resulting models include a table showing the importance of each variable or feature for predicting. I compared the tables to the feature sets produced by my algorithm; the results appear in Tables 3 - 6. Each table shows the importance of each feature as a percentage, with importance decreasing from left to right. The highlighted columns indicate the features found by the Few algorithm. I also calculated the accuracy of the two models on all features with at least 50% importance; the results are shown in Figure 4.

Table 3: Importance of features according to the rpart2 model when trained on F_{all}. The highlighted columns indicate the set of features discovered by the algorithm.

P	C	A	Ed	**SF1**	x	CO	SF3	K	**X**	Ec	Ex	S	**R**	**SF2**	SF4
100	97.044	96.44	96.44	**95.92**	95.349	93.931	93.931	93.438	**57.304**	42.378	0	0	**0**	**0**	0

Table 4: Importance of features according to the rpart2 model when trained on F_{min}. The highlighted columns indicate the set of features discovered by the algorithm.

x	**SF1**	**X**	A	SF3	Ec	**P**	**SF2**	SF4	Ex
100	**84.985**	**82.711**	68.569	65.967	65.956	**63.134**	**33.766**	12.898	0

Table 5: Importance of features according to the treebag model when trained on F_{all}. The highlighted columns indicate the set of features discovered by the algorithm.

CO	K	SF3	x	SF1	**A**	C	Ed	**P**	Ec	X	**R**	**SF2**	**SF4**	S	Ex
100	98.949	97.413	90.144	88.962	**83.928**	79.66	75.207	**72.12**	48.505	42.972	**22.151**	**16.414**	**9.6494**	4.0059	0

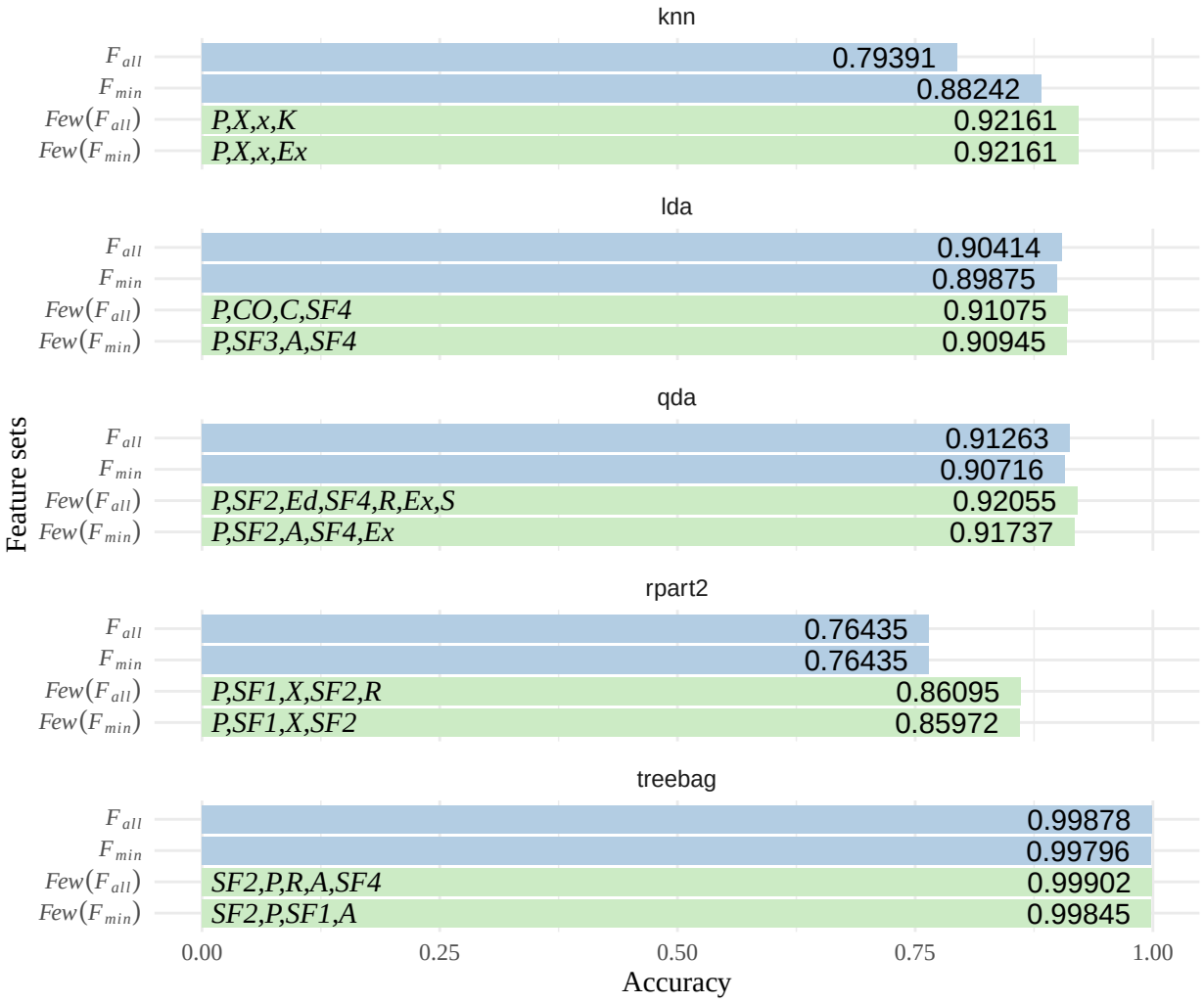

Figure 3: Accuracy of each model on the two feature sets and on the minimal sets produced by the algorithm (denoted by $Few(F_{all})$ and $Few(F_{min})$). The feature sets discovered by the algorithm are shown at the base of the corresponding bar.

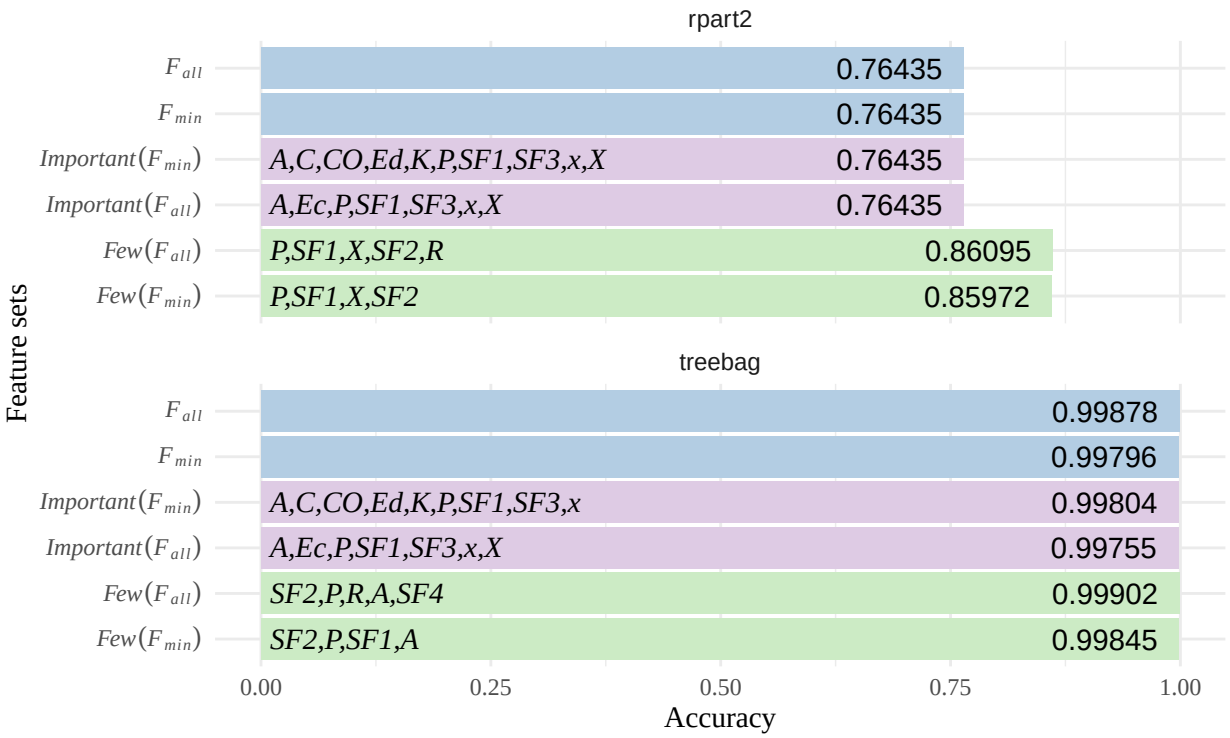

Figure 4: Accuracies of different sets of features on the rpart2 and treebag models. The bars marked *Important* show the accuracy on the set of features that has at least 50% importance according to the respective model.

Table 6: Importance of features according to the treebag model when trained on F_{all}. The highlighted columns indicate the set of features discovered by the algorithm.

x	**SF1**	X	**A**	SF3	Ec	**P**	**SF2**	SF4	Ex
100	**84.985**	82.711	**68.569**	65.967	65.956	**63.134**	**33.766**	12.898	0

4 Results

The minimal feature set F_{min} reduced accuracy only slightly (a maximum loss of 0.6%) compared to using the full feature set F_{all}. It even improved accuracy by 11.1% in the `knn` model. The feature sets found by the algorithm improved accuracy on the training set in every model compared to F_{all}, though in three of the models (`lda`, `qda`, and `treebag`) the improvement was by less than 1%. Furthermore, the largest feature set found by the algorithm had only 7 features and all other discovered feature sets had either 4 or 5 features. This is a significant reduction in size from the 10 features in F_{min}.

On the other hand, the features reported as important by the `rpart2` and `treebag` models did not improve accuracy. Also, the features found by the algorithm showed no particular relation to the important features. In fact, two of the features found by the algorithm for `rpart2` have an importance of zero (Table 3).

With the exception of the `treebag` model, the accuracy results on the testing set are very similar to the accuracy results on the training set (Figure 5). The accuracy results for the `treebag` model were very high on the training set, suggesting that it may have been overtrained. The accuracies for the feature sets found by the algorithm were also lower on the testing set than the accuracies for the original feature sets, which may also indicate overtraining.

5 Conclusions

Judicious feature selection can clearly improve the accuracy of some models. The algorithm improved the accuracy of each model and simultaneously reduced the required number of features significantly. It has one major disadvantage: it may increase the training time significantly. In the worst case, where every additional feature increases accuracy, the training time is multiplied by a factor of $N^2/2$, where N is the number of features. For the models chosen, the actual training time was multiplied by a factor between $4N$ and $7N$, compared to the training time on a single feature set. Training for the `lda` and `qda` models was very fast, roughly 10 seconds per feature set on the training set, so the time taken by the algorithm was not prohibitive. By contrast, training for the `treebag` model took 4-5 minutes per feature set, so the algorithm took several hours to complete. Some of the other random forest models, such as `rf` and `Rborist`, took substantially longer, which is why they were not chosen for this report. The size of the dataset and the speed of training a particular model on a feature set are therefore useful guides in deciding whether the algorithm is worth applying.

Bibliography

Koklu, Murat, and Ilker Ali Ozkan. 2020. "Multiclass Classification of Dry Beans Using Computer Vision and Machine Learning Techniques." *Computers and Electronics in Agriculture* 174 (July): 105507. https://doi.org/10.1016/j.compag.2020.105507.

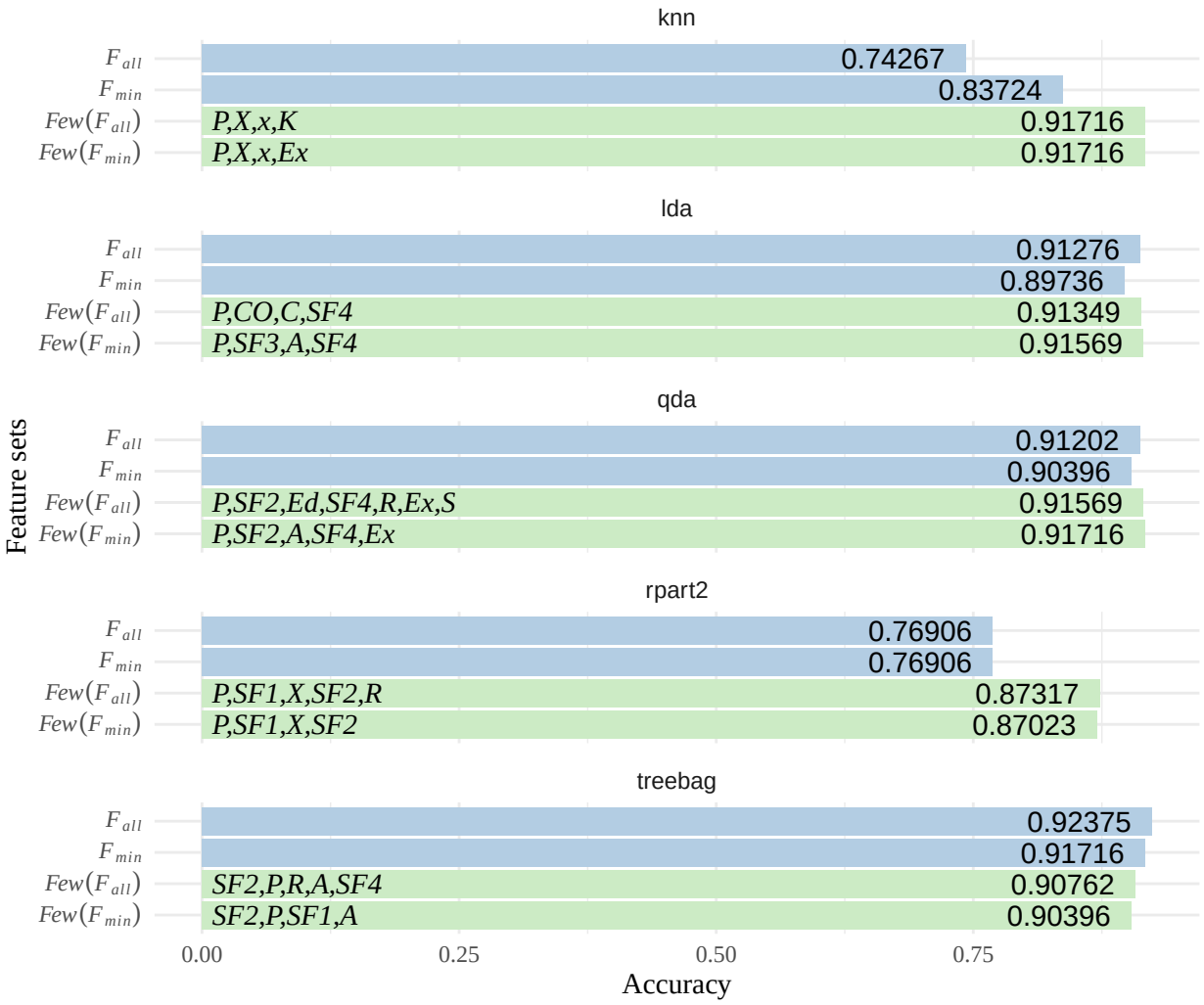

Figure 5: Accuracy of each model and feature set on the testing set.