HarvardX Capstone Project: Kanji Grade Finder

Ayomide Bamgbose

04/01/2021

Executive Summary

The purpose of this report is to implement a machine learning model on a given dataset. The goal of this report is to create a machine learning model that will predict the JLPT grade of a given kanji. It will strive to provide an aid for learners of the Japanese language. The `random forest` model was selected to train the dataset, and gave an accuracy of `0.636`. This accuracy is likely due to the small size of the dataset, and expanding the given dataset to include JLPT-specific vocabulary would most likely increase the model accuracy without causing over-training to occur.

Introduction

The Japanese language is composed of three alphabets: hiragana (平仮名), katakana (カタカナ), and kanji (漢字). The kanji alphabet is composed of originally Chinese characters that have been adjusted and implemented into the Japanese language over time. For individuals hoping to learn the Japanese language, this kanji system is often the most difficult part of their experience. This is primarily due to the fact that there are thousands of kanji, and each kanji often has different meanings and readings ("readings" referring to the way the kanji is read in a certain circumstance). Due to this, a list of 2136 kanji (dubbed, "Joyo kanji") was created by the Japanese Ministry of Education. This list contained the kanji that were most likely to be seen on a day-to-day basis (in Japan). It is often said that these are the "only" kanji a learner of Japanese needs to know in order to be proficient in the language. Each kanji in this list is also taught to Japanese individuals throughout their elementary to highschool education. In other words, each kanji in this list is attached to a specific grade of study.

For learners of Japanese, there is a test called the *Nihongo Nouryoku Shiken* (本語能力試験, Japanese Language Proficiency Test, JLPT) that is used to test a learner's proficiency in Japanese. There are 5 levels to this test, and each level contains a group of kanji. It is often difficult for learners to determine whether or not a given kanji is at their preferred level of study by just looking at its grade (since this grade is for the Japanese education system). However, by knowing the JLPT grade of a given kanji, the learning process can be simplified. Since the majority of learners of Japanese will take the JLPT, it is

very important that they know which kanji to learn (according to their level). This report will implement a machine learning model that will predict the JLPT grade of a given kanji.

Overview

In order to implement a machine learning model, the required data will first need to be selected. Following this, data visualization and exploration will be conducted in order to gain an understanding of the behavior and characteristics of this data. Various machine learning models will then be tested and compared, and a final model will be selected. The highest JLPT level is 1 (often referred to as "N1"), and the lowest JLPT level is 5 (often referred to as "N5").

Dataset

Three datasets will be used for this project:

1. List of joyo kanji
2. List of radicals by stroke count
3. Kanji dataset

■List of joyo kanji This dataset contains all 2,136 Joyo kanji and is defined with the following variables:

- `kanjiID`: the ID of the given kanji
- `kanji`: the kanji (in Japanese form)
- `old`: the old version of the kanji
- `radical`: the radical of the kanji
- `kanji_stroke_count`: the number of strokes in the kanji
- `grade`: the grade of the kanji with respect to the Japanese education system (e.g., "1" is grade 1 and "S" is secondary school (grade 7-9))
- `year_added`: the year the given kanji was added to the Joyo kanji list
- `meaning`: the menaing of the kanji
- `readings`: the reading(s) of the kanji

The structure of this dataset is given as follows:

```
## 'data.frame':    2136 obs. of  9 variables:
##  $ kanjiID           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ kanji             : chr  "亜" "哀" "挨" "愛" ...
##  $ old               : chr  "亞" "" "" "" ...
##  $ radical           : chr  "二" "口" "手" "心" ...
##  $ kanji_stroke_count: int  7 9 10 13 17 11 12 5 6 8 ...
```

```
##  $ grade             : chr  "S" "S" "S" "4" ...
##  $ year_added        : num  1946 1946 2010 1946 2010 ...
##  $ meaning           : chr  "sub-" "pathetic" "push open" "love" ...
##  $ readings          : chr  "ア a" "アイ、あわ-れ、あわ-れむ ai, awa-re, awa-remu" "アイ
ai" "アイ ai" ...
```

Note: "Strokes" refer to the number of times you must lift up the pen/pencil when writing a kanji or radical. For example:

```
##   kanji kanji_stroke_count
## 1    亜                  7
```

There is a specific system for counting the strokes of a kanji, but it will not be discussed in this report.

■List of radicals by stroke count This dataset contains 8 variables:

- `radicalID`: the numerical ID of the radical
- `radical`: the radical (in Japanese)
- `radical_stroke_count`: the number of strokes in the radical
- `meaning_reading`: the meaning and reading of the radical
- `freq`: the frequency of the radical (from the 47,035 characters in the Chinese language)
- `joyo_freq`: the frequency of the radical (from the 2,136 Joyo kanji)
- `examples`: examples of some kanji that use this radical
- `group`: "Top 25%" means that this radical represents 25% of Jōyō kanji. "Top 50%" means that this radical plus the "Top 25%" represent 50% of Jōyō kanji. "Top 75%" means that this radical plus the "Top 50%" represent 75% of Jōyō kanji

```
## 'data.frame':    214 obs. of  8 variables:
##  $ radicalID          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ radical            : chr  "一" "丨" "丶" "丿" ...
##  $ radical_stroke_count: int  1 1 1 1 1 1 2 2 2 2 ...
##  $ meaning_reading    : chr  "one (いち, ichi, 一)" "line, stick (ぼう, bō, 棒)" "dot (て
ん, ten, 点)" "bend, possessive particle no (の, no, ノ)" ...
##  $ freq               : num  42 21 10 33 42 19 29 38 794 52 ...
##  $ joyo_freq          : num  50 0 0 0 0 0 0 0 83 0 ...
##  $ examples           : chr  "七三丈不丘世" "中" "丸主" "久之乎" ...
##  $ group              : chr  "" "" "" "" ...
```

Note: "Radicals" are the smaller parts that make up a kanji. For example:

```
##   radical     examples
```

```
## 1        一 七三丈不丘世
```

■Kanji Dataset This dataset contains the following variables:

- `ID`: the ID of the given kanji
- `kanji`: the kanji (in Japanese form)
- `kanji_stroke_count`: the number of strokes in the kanji
- `grade`: the grade of the kanji with respect to the Japanese education system (e.g., "1" is grade 1 and "S" is secondary school (grade 7-9))
- `classification`: "Rikusho Bunrui" kanji classification
- `JLPT_grade`: the grade of the kanji with respect to the Japanese Language Proficiency Test
  - Note: a JLPT grade of "0" indicates the fact that the kanji is not ranked in the JLPT 1-5 levels
  - The highest (i.e., most difficult) JLPT level is 1, and the lowest (i.e., easiest) JLPT level is 5
- `radical`: the radical of the kanji
- `radical_freq`: the frequency of the radical (from the 2,136 Joyo kanji)
- `n_on_readings`: number of onyomi readings
- `n_on_meanings`: number of meanings for onyomi readings
- `n_kun_meanings`: number of meanings for kunyomi readings
- `year_added2`: the year the given kanji was added to the Joyo kanji list
- `kanji_freq_proper`: total kanji frequency from the "Mainichi Shinbun" including proper nouns
- `kanji_freq`: total kanji frequency from the "Mainichi Shinbun" *not* including proper nouns
- `symmetry`: lexical symmetry of the kanji from left to right
  - "S": symmetric
  - "R": right-symmetric
  - "L": left-symmetric

```
## 'data.frame':    2136 obs. of  15 variables:
##  $ ID                : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ kanji             : chr  "亜" "哀" "挨" "愛" ...
##  $ kanji_stroke_count: num  7 9 10 13 17 11 12 5 6 8 ...
##  $ grade             : num  7 7 7 4 7 3 7 5 7 7 ...
##  $ classification    : chr  "象形_Pictographic" "会意_Com_Ideographic" "形声_Phonetic" "
会意_Com_Ideographic" ...
##  $ JLPT_grade        : num  1 1 0 3 0 4 1 2 1 0 ...
##  $ radical           : chr  "Ni" "Kuchi_Kuchihen" "Tehen" "Kokoro_Risshinben_Shitagokoro" ...
##  $ radical_freq      : num  6 70 89 76 38 76 89 50 89 36 ...
##  $ n_on_readings     : num  1 1 1 1 1 2 1 1 0 0 ...
##  $ n_on_meanings     : num  5 3 1 3 2 5 6 4 0 0 ...
```

```
##  $ n_kun_meanings    : num  0 8 0 0 0 17 6 0 8 1 ...
##  $ year_added2       : num  1981 1981 2010 1981 2010 ...
##  $ kanji_freq_proper : num  13829 4792 324 94602 214 ...
##  $ kanji_freq        : num  1457 4651 324 50443 214 ...
##  $ symmetry          : chr  "S" "R" "" "R" ...
```

Note: *onyomi* and *kunyomi* readings refer to the Chinese and Japanese readings of the kanji, respectively.

Methods

Applying the three datasets given above, they must be merged into one cohesive dataset and cleaned. From this new dataset, training and test sets will also be created.

Data Cleaning

In order to clean and merge the data, each dataset must be cleaned separately in advance. Since all datasets have overlapping/identical data, some duplicate columns can be removed.

The following columns are removed from each dataset:

```
## $radical_df
## [1] "radicalID"       "meaning_reading" "examples"        "freq"
##
## $kanji_df
## [1] "readings"           "meaning"            "kanjiID"
## [4] "grade"              "kanji_stroke_count"
##
## $kanji_df2
## [1] "year_added2"
```

In addition to this, the following columns are converted into factors:

```
## $radical_df
## [1] "group"
##
## $kanji_df
## [1] "year_added"
##
## $kanji_df2
## [1] "grade"          "classification" "JLPT_grade"     "radical"
## [5] "symmetry"
```

As soon as each dataset has been tidied separately, they are merged together with the `left_join()` function. The final dataset is thus defined as:

```
## 'data.frame':    2136 obs. of  20 variables:
##  $ ID                : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ kanji             : chr  "亜" "哀" "挨" "愛" ...
##  $ kanji_stroke_count: num  7 9 10 13 17 11 12 5 6 8 ...
##  $ grade             : Factor w/ 7 levels "1","2","3","4",..: 7 7 7 4 7 3 7 5 7 7 ...
##  $ classification    : Factor w/ 6 levels "仮借_Loan","会意_Com_Ideographic",..: 6 2 4 2 4 4 4 2 4
##  $ JLPT_grade        : Factor w/ 6 levels "0","1","2","3",..: 2 2 1 4 1 5 2 3 2 1 ...
##  $ radical.x         : Factor w/ 204 levels "Aka","Akubi",..: 124 95 172 87 125 87 172 182 172 19
##  $ radical_freq      : num  6 70 89 76 38 76 89 50 89 36 ...
##  $ n_on_readings     : num  1 1 1 1 1 2 1 1 0 0 ...
##  $ n_on_meanings     : num  5 3 1 3 2 5 6 4 0 0 ...
##  $ n_kun_meanings    : num  0 8 0 0 0 17 6 0 8 1 ...
##  $ kanji_freq_proper : num  13829 4792 324 94602 214 ...
##  $ kanji_freq        : num  1457 4651 324 50443 214 ...
##  $ symmetry          : Factor w/ 4 levels "","P","R","S": 4 3 1 3 1 3 1 4 1 4 ...
##  $ old               : chr  "亞" "" "" "" ...
##  $ radical           : Factor w/ 202 levels "","一","丨","丶",..: 8 31 64 61 72 61 64 33 64 40 ...
##  $ year_added        : Factor w/ 3 levels "1946","1981",..: 1 1 3 1 3 1 1 1 1 3 ...
##  $ joyo_freq         : num  0 100 68 67 51 67 68 42 68 37 ...
##  $ group             : Factor w/ 4 levels "","Top 25%","Top 50%",..: 1 2 2 2 3 2 2 3 2 3 ...
##  $ old_logical       : logi  TRUE FALSE FALSE FALSE FALSE TRUE ...
```

■Create the training and test sets After setting the seed to **1**, the training set is defined as 80% of the data and the test set is defined as 20% of the data. This value is selected because the dataset is relatively small (there are only 2136 observations).

```
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = df$JLPT_grade, times = 1, p = 0.2, list = FALSE)

kanji_train <- df[-test_index, ]
kanji_test <- df[test_index, ]
```

Data Exploration

The main characteristics and behaviour of the `kanji_train` dataset will be explored in this section. The `kanji_train` dataset will be analyzed because the machine learning model must be trained with this

data.

■most_common() function This function will be used in subsequent data analysis. It takes the form `function(x,y)`, where `x` is a vector containing all data and `y` is a vector containing the unique values (e.g., levels) of the vector `x`. The formula is shown here:

```
most_common <- function(x, y){
  count <- vector(mode = "numeric", length = length(y))
  for(i in seq(1, length(y), 1)){
    count[i] <- sum(x == y[i])
  }
  return(y[which.max(count)])
}
```

For example, the most common grade in the dataset can be determined by:

```
most_common(kanji_train$grade, levels(kanji_train$grade)) # = 7
```

Applying this function to the entire `kanji_train` dataset, the most common (for categorical data) and average (for numerical data) for each predictor in the dataset is shown in the table below. This table is created by grouping the `kanji_train` dataset by each JLPT grade:

```
## # A tibble: 6 x 7
##   JLPT_grade grade classification    radical avg_strokes percent_old on_meanings
##   <fct>      <chr> <chr>             <chr>         <dbl>       <dbl>       <dbl>
## 1 0          7     形声_Phonetic     ""             11.6       0.0231       2.23
## 2 1          7     形声_Phonetic     "手"           11.5       0.201        3.40
## 3 2          7     形声_Phonetic     "水"           10.1       0.184        3.62
## 4 3          3     形声_Phonetic     "人"           10         0.196        5.01
## 5 4          2     形声_Phonetic     "人"            8.55      0.173        4.62
## 6 5          1     象形_Pictographic "十"            5.98      0.0781       3.81
```

From the table above, it follows that the JLPT grades higher than level 3 are most often composed of kanji in grade 7 (i.e., secondary school). The most common Kanji classification amongst all JLPT grades is the `形声_Phonetic` classification. The average number of strokes for a kanji (`avg_strokes`) tends to increase as the JLPT level increases. The level 1 JLPT level has the highest percentage of old kanji versions (`old_percentage`).

```
## # A tibble: 6 x 4
##   JLPT_grade kun_meanings on_readings radical_freq
```

```
##   <fct>             <dbl>       <dbl>        <dbl>
## 1 0                  1.88        0.9          42.0
## 2 1                  2.95        1.03         43.1
## 3 2                  4.50        1.07         44.4
## 4 3                  7.25        1.22         41.8
## 5 4                  7.80        1.35         32.5
## 6 5                  9.33        1.52         25.4
```

From the table above, it follws that the lower JLPT levels tend to have a higher average number of *onyomi* readings (`on_meanings`). In a similar manner, the JLPT level 5 grade has the highest average number of *kunyomi* readings (`kun_meanings`). The average radical frequency (`radical_freq`) does not vary much across each JLPT level, but the JLPT levels 4 and 5 tend to be composed of lower frequency radicals.

The table below shows that the higher the JLPT level of the kanji, the lower the average frequency (`avg_freq` and `avg_freq_proper`) of the kanji. The most common symmetry is `S` and most common year added is `1946`.

```
## # A tibble: 6 x 5
##   JLPT_grade avg_freq avg_freq_proper symmetry year_added
##   <fct>         <dbl>           <dbl> <chr>    <chr>
## 1 0             1797.           3437. ""       2010
## 2 1            19352.          24535. "S"      1946
## 3 2            47406.          57992. "S"      1946
## 4 3           132234.         146194. "S"      1946
## 5 4           221120.         251162. "P"      1946
## 6 5           381376          475314  "R"      1946
```

17.7% of the kanji in the dataset have an "old" version, however there is a very small difference in the stroke count of kanjis with and without old versions:

```
## # A tibble: 12 x 4
## # Groups:   JLPT_grade [6]
##    JLPT_grade old_logical     n avg_stroke_count
##    <fct>      <lgl>       <int>            <dbl>
##  1 0          FALSE         127             11.6
##  2 0          TRUE            3             11.7
##  3 1          FALSE         631             11.3
##  4 1          TRUE          159             12.3
##  5 2          FALSE         240             10.0
##  6 2          TRUE           54             10.2
```

```
##  7 3          FALSE         238             9.86
##  8 3          TRUE           58            10.6
##  9 4          FALSE         110             8.41
## 10 4          TRUE           23             9.22
## 11 5          FALSE          59             5.90
## 12 5          TRUE            5             7
```

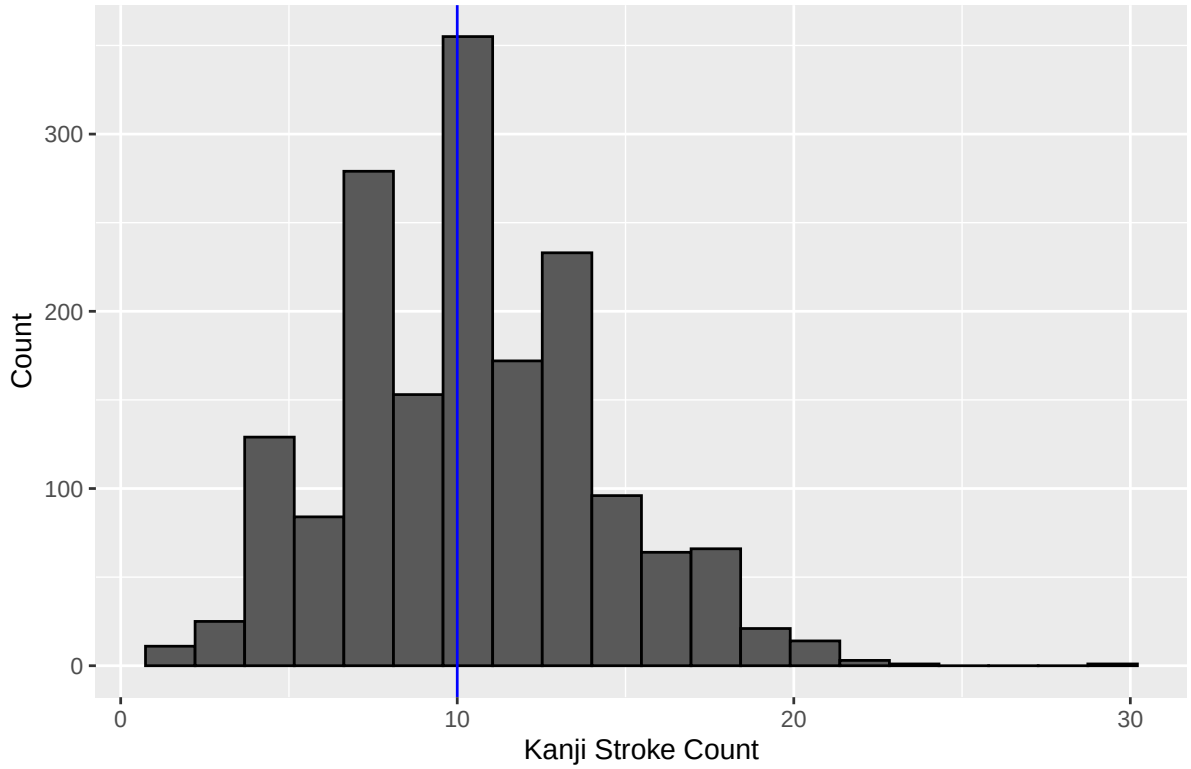
In addition, there does not seem to be a significant relationship between the JLPT grade of a kanji and the most common group of the radicals in that grade:

```
## # A tibble: 6 x 2
##   JLPT_grade group
##   <fct>      <chr>
## 1 0          ""
## 2 1          ""
## 3 2          ""
## 4 3          ""
## 5 4          ""
## 6 5          ""
```

Data Visualization

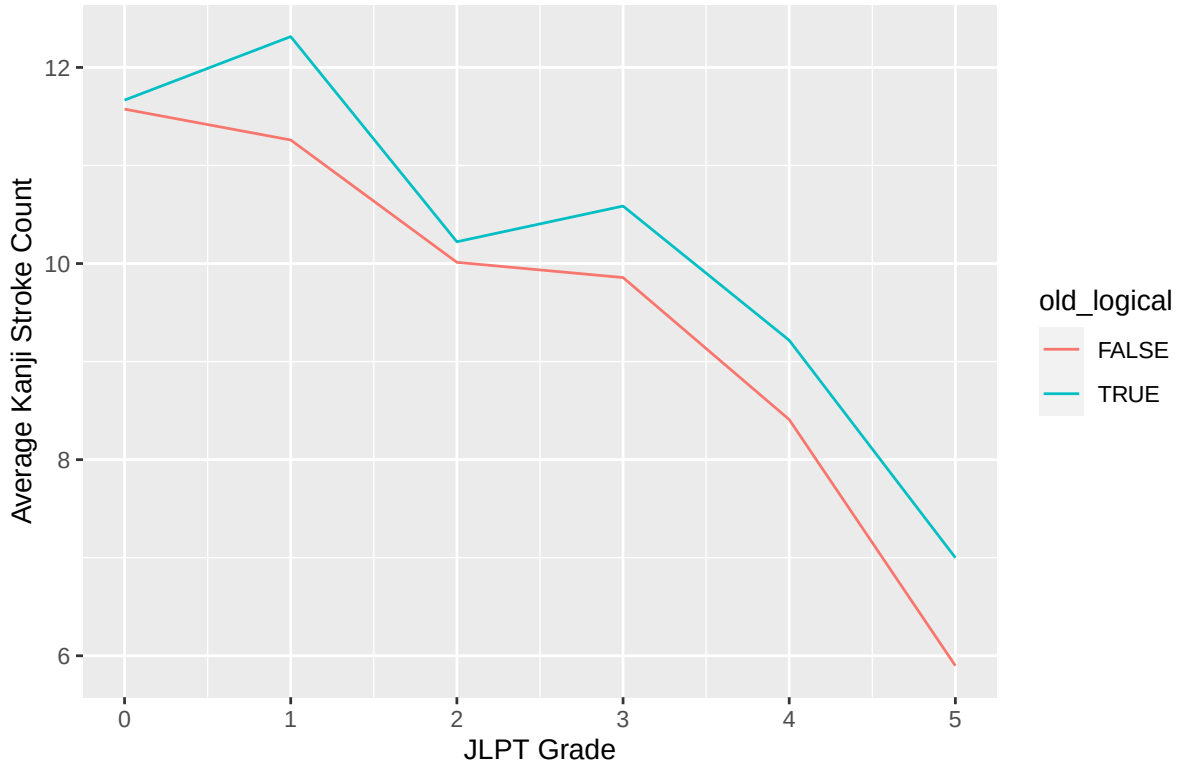
Figure 1, below, shows that the `kanji_stroke_count` predictors is normally distributed with an median of 10 strokes:

Figure 1. Kanji Stroke Count Distribution



Calculating the average number of strokes (per kanji) over each JLPT grade, the following can be shown:

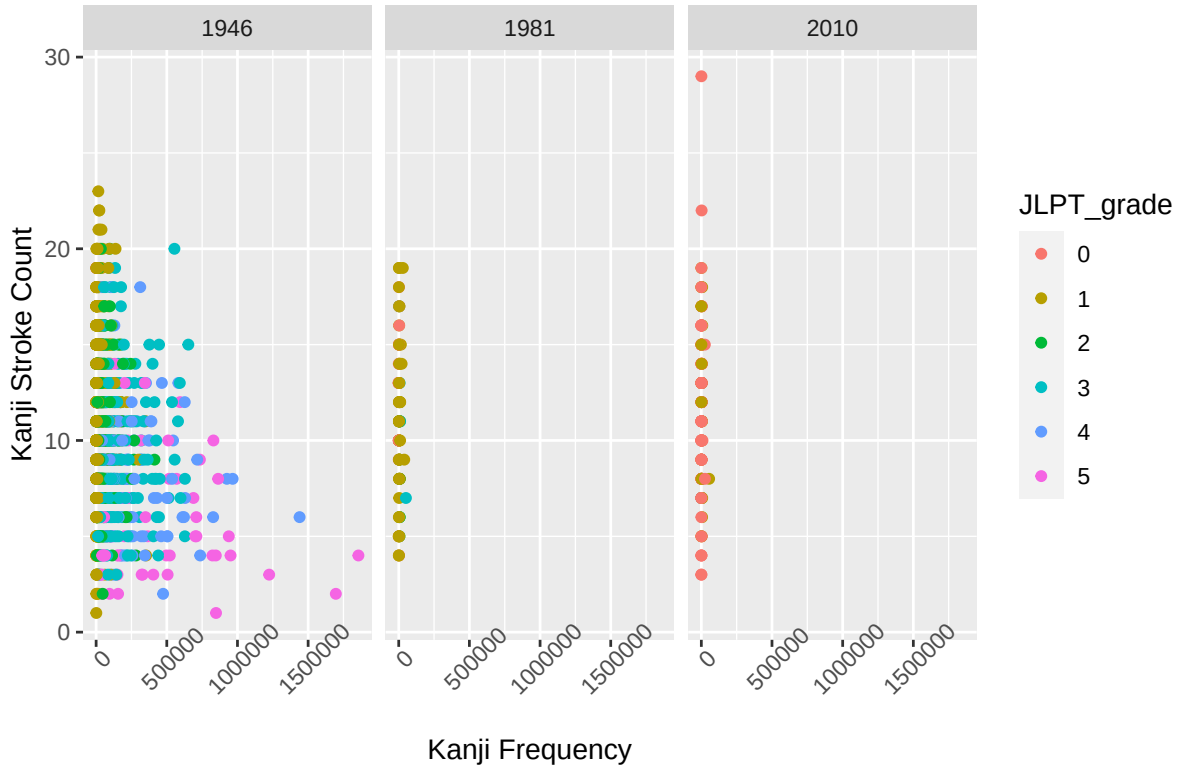
Figure 2. Kanji Stroke Count vs JLPT Grade



This figure shows a trend of a decline in the average number of strokes a kanji has versus its JLPT grade. The higher the JLPT grade, the higher the number of strokes the kanji has. In addition, kanjis without and old version tend to have a lower number of strokes (in comparison to kanjis with an old version). This could imply that if a kanji has an old version, it is likely that it has a higher number of strokes.

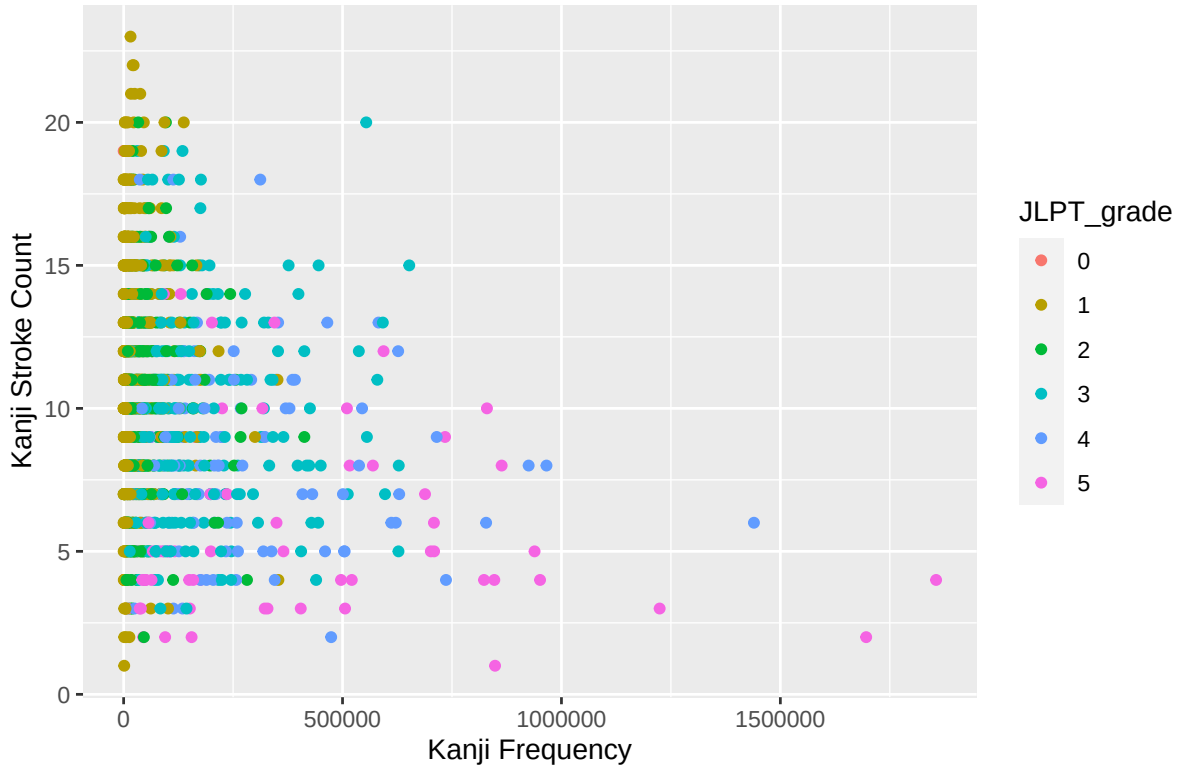
From the figure below, it is clear that almost all of the kanji in the `kanji_train` dataset were added to the Joyo kanji in 1946:

Figure 3. Kanji Stroke Count vs Frequency (by Year Added)



Re-plotting this figure so that only the kanji added in 1946 are selected:

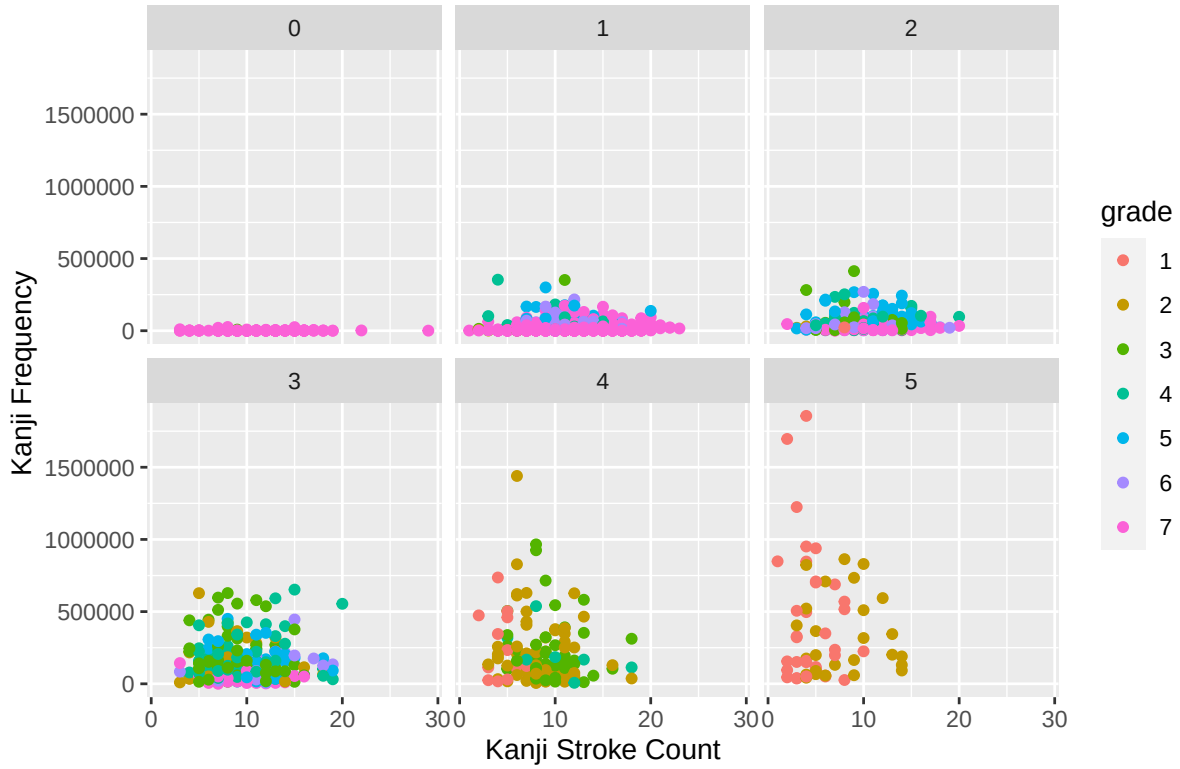
Figure 4. Kanji Stroke Count vs Frequency (in 1946)



From the figure above, it is clear that the higher JLPT grade (i.e., level 1 or 2) kanji tend to be made up of low-frequency kanjis. Also, low JLPT grade (i.e., level 4 or 5) kanji tend to be made up of kanjis with low stroke counts.

Figure 5, below, shows that there is a distinct correlation between the `JLPT_grade` of a kanji and its `grade`:

Figure 5. Kanji Frequency vs Stroke Count (by JLPT Grade)



Compiling the plots above into a table, the correlation between the `JLPT_grade` of a kanji and its `grade` can be summarised:

```
##   JLPT_grade           grades
## 1          0                7
## 2          1          4, 5, 7
## 3          2       3, 4, 5, 7
## 4          3 2, 3, 4, 5, 6, 7
## 5          4       1, 2, 3, 4
## 6          5             1, 2
```

From the table above, it follows that the JLPT level 5 is composed of grade 1 and 2 kanji. The JLPT levels 0 to 2 are mainly composed of grade 4-7 kanji.

Analysis

Since the `JLPT_grade` predictor is a categorical variable, we can only use machine learning models that can work with categorical outcomes. Due to the structure of the predictors in the `kanji_train` dataset, four machine learning models will be used in this report:

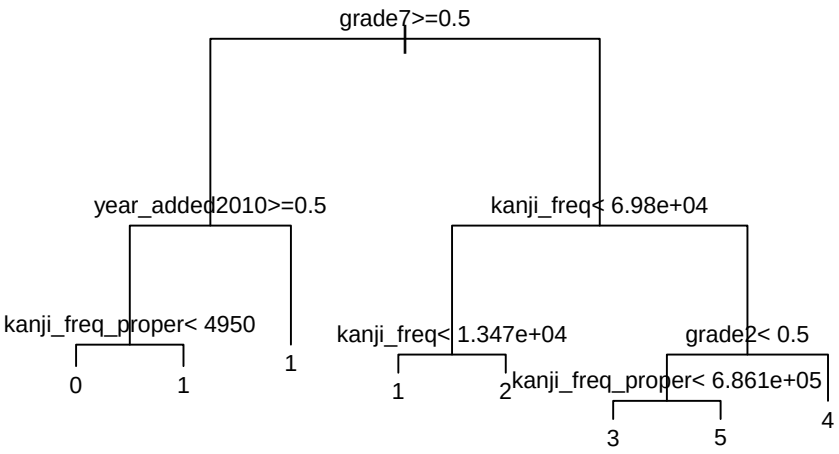
1. Decision trees (`rpart`)
2. Random forest (`rf`)
3. K-Nearest Neighbors (`knn`)
4. Linear Discriminant Analysis (`lda`)
5. Ensemble

Since the `kanji` and `ID` predictors contain only unique values, they will not be useful as predictors in any machine learning algorithms. A `predictors` dataset is taken from the `kanji_train` dataset, and is defined with the following variables:

```
## 'data.frame':    1707 obs. of  17 variables:
##  $ kanji_stroke_count: num  7 9 10 13 11 12 5 6 8 12 ...
##  $ grade             : Factor w/ 7 levels "1","2","3","4",..: 7 7 7 4 3 7 5 7 7 7 ...
##  $ classification    : Factor w/ 6 levels "仮借_Loan","会意_Com_Ideographic",..: 6 2 4 2 4 4 2 4 2
##  $ JLPT_grade        : Factor w/ 6 levels "0","1","2","3",..: 2 2 1 4 5 2 3 2 1 2 ...
##  $ radical.x         : Factor w/ 204 levels "Aka","Akubi",..: 124 95 172 87 87 172 182 172 191 20
##  $ radical_freq      : num  6 70 89 76 76 89 50 89 36 16 ...
##  $ n_on_readings     : num  1 1 1 1 2 1 1 0 0 0 ...
##  $ n_on_meanings     : num  5 3 1 3 5 6 4 0 0 0 ...
##  $ n_kun_meanings    : num  0 8 0 0 17 6 0 8 1 2 ...
##  $ kanji_freq_proper : num  13829 4792 324 94602 91570 ...
##  $ kanji_freq        : num  1457 4651 324 50443 91516 ...
##  $ symmetry          : Factor w/ 4 levels "","P","R","S": 4 3 1 3 3 1 4 1 4 2 ...
##  $ radical           : Factor w/ 202 levels "","一","丨","丶",..: 8 31 64 61 61 64 33 64 40 46 ...
##  $ year_added        : Factor w/ 3 levels "1946","1981",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ joyo_freq         : num  0 100 68 67 67 68 42 68 37 0 ...
##  $ group             : Factor w/ 4 levels "","Top 25%","Top 50%",..: 1 2 2 2 2 2 3 2 3 4 ...
##  $ old_logical       : logi  TRUE FALSE FALSE FALSE TRUE FALSE ...
```

Decision tree, `rpart`

Training a decision tree (`rpart`) model on the `predictors` dataset, the following decision tree is created with tuning parameter `cp = 0.01`:



Using the `varImp()` function, the four most important predictors are: `kanji_freq_proper`, `kanji_freq`, `grade` and `year_added`. The accuracy of this model is given by:

```
rpart_cm <- confusionMatrix(predict(rpart_fit, kanji_test), kanji_test$JLPT_grade)
rpart_cm$overall["Accuracy"]
```

```
## Accuracy
##    0.587
```

Random forest, `rf`

Training a random forest model with the `randomForest()` function, the tuning parameter `mtry = 3` is selected. The five most important predictors are: `kanji_stroke_count`, `grade`, `kanji_freq_proper`, `kanji_freq`, and `radical_freq`. The accuracy of this model is given by:

```
rforest_cm <- confusionMatrix(predict(rforest_fit, kanji_test), kanji_test$JLPT_grade)
rforest_cm$overall["Accuracy"]
```

```
## Accuracy
```

```
##    0.636
```

K-Nearest Neighbors, `knn`

The k-nearest neighbors model is trained with 10-fold cross validation for `k = seq(20, 30, 2)`. The final model uses `k = 28`. The five most important predictors for this model are: `kanji_freq_proper`, `kanji_freq`, `grade`, `year_added`, and `symmetry`. The accuracy of the model is:

```
knn_cm <- confusionMatrix(predict(knn_fit, kanji_test), kanji_test$JLPT_grade)
knn_cm$overall["Accuracy"]
```

```
## Accuracy
##    0.543
```

Linear Discriminant Analysis, `lda`

Since `lda` cannot be used with predictors that are factors, the un-factorized, character, and numerical predictors are used instead:

The variable importance cannot be determined with linear discriminant analysis, so the accuracy will be calculated instead:

```
lda_cm <- confusionMatrix(predict(lda_fit, kanji_test), kanji_test$JLPT_grade)
lda_cm$overall["Accuracy"]
```

```
## Accuracy
##    0.566
```

Note: `qda` (Quadrant Discriminant Analysis) is not used in this project because there are too many levels in each predictor. Attempting to train a `qda` machine learning model results in rank deficiency issues.

Ensemble Model

This model is composed of a combination of the four previous models. Predictions for this model are made by choosing the most common prediction (with the `most_common(x,y)` function) out of all four models.

```
# combine predictions of all models
predictions <- data.frame(rpart = predict(rpart_fit, kanji_test),
                          rforest = predict(rforest_fit, kanji_test),
```

```
                          knn = predict(knn_fit, kanji_test),
                          lda = predict(lda_fit, kanji_test))
str(predictions)
```

```
## 'data.frame':    429 obs. of  4 variables:
##  $ rpart  : Factor w/ 6 levels "0","1","2","3",..: 1 4 2 3 2 2 2 2 2 3 ...
##  $ rforest: Factor w/ 6 levels "0","1","2","3",..: 1 4 2 4 1 2 2 2 2 3 ...
##  $ knn    : Factor w/ 6 levels "0","1","2","3",..: 1 4 2 3 1 2 2 2 4 3 ...
##  $ lda    : Factor w/ 6 levels "0","1","2","3",..: 2 4 2 4 2 2 2 2 2 4 ...
```

```
rows <- seq(1:nrow(predictions))
outcomes <- levels(kanji_test$JLPT_grade)

# create ensemble prediction by selecting the most common prediction acrosss all models
ensemble_pred <- sapply(rows, function(r){
  most_common(predictions[r,], outcomes)
})
```

For example, the ensemble predictions for the first three kanji are:

```
##   rpart rforest knn lda ensemble
## 1     0       0   0   1        0
## 2     3       3   3   3        3
## 3     1       1   1   1        1
```

Results

Compiling the accuracy of each model in the previous section, the following table can be created:

```
##           model accuracy
## 1 decision tree    0.587
## 2 random forest    0.636
## 3           knn    0.543
## 4           lda    0.566
## 5      ensemble    0.608
```

From this table, it is clear that the `random forest` and `ensemble` models have the highest accuracy. Since the random forest model has the highest accuracy, it will be used as the final model. The accuracy of the selected model is:

```r
## Selected: randomForest model
confusionMatrix(predict(rforest_fit, kanji_test), kanji_test$JLPT_grade)$overall["Accuracy"]
```

```
## Accuracy
##    0.634
```

Conclusion

The `random forest` model was selected to train the `kanji_train` dataset, and gave an accuracy of `0.636`. The most important predictors are `kanji_freq`, `kanji_freq_proper`, `grade`, and `year_added`. Since the `kanji_train` dataset is quite small, it would be difficult to fit an accurate algorithm to it without over-training. A possible solution to this problem would be to include the words (in kanji form) that are commonly found in each JLPT level. This would greatly expand the amount of available data for this machine learning challenge, and would most likely allow for an increase in the model accuracy without causing over-training. Through further improvement, this model can be used as a simple guide for learners of Japanese by ensuring that the kanji they are learning is appropriate for their level. In the future, this dataset would be expanded to include JLPT-specific vocabulary, and additional machine learning models (e.g., matrix factorization) would be tested.

References

[1] Tamaoka, K., Makioka, S., Sanders, S. & Verdonschot, R.G. (2017). www.kanjidatabase.com