Problem set 10

2025-04-20

The data for this problem set is provided by this link: `https://github.com/dmcable/BIOSTAT620/raw/refs/he`

Read this object into R. For example, you can use:

The object is a list with two components `dat$train` and `dat$test`. Use the data in `dat$train` to develop a machine learning algorithms to predict the labels for the images in the `dat$test$images` component.

Save the your predicted labels in an object called `digit_predictions`. This should be a vector of integers with length `nrow(dat$test$images)`. It is important that the `digit_predictions` is ordered to match the rows of `dat$test$images`.

Save the object to a file called `digit_predictions.rds` using:

You will submit:

1. The file `digit_predictions.rds`
2. A quarto file that reproduces your analysis and provides brief explanations for your choices.

If your code reproduces the result, your grade will be your accuracy rounded up the closest integer. So, for example, if your accuracy is .993 your grade will be 100%. Depending on the distribution of accuracy values, the teaching staff may issue an update about the grading system used.

You will have one opportunities to redo your predictions after you see your accuracy from your first submission.

Load Data

```
# Load the dataset
mnist <- readRDS("pset-10-mnist.rds")
X_train <- mnist$train$images
y_train <- as.factor(mnist$train$labels)
X_test <- mnist$test$images
```

Model Fitting: Random Forest (Ranger)

```
# Combine into training dataframe
train_df <- as.data.frame(X_train)
train_df$label <- y_train

# Train ranger model (fast implementation of Random Forest)
set.seed(42)
rf_model <- ranger(
  formula = label ~ .,
  data = train_df,
  num.trees = 100,
  importance = "impurity",
  probability = FALSE,
  num.threads = parallel::detectCores()
)
```

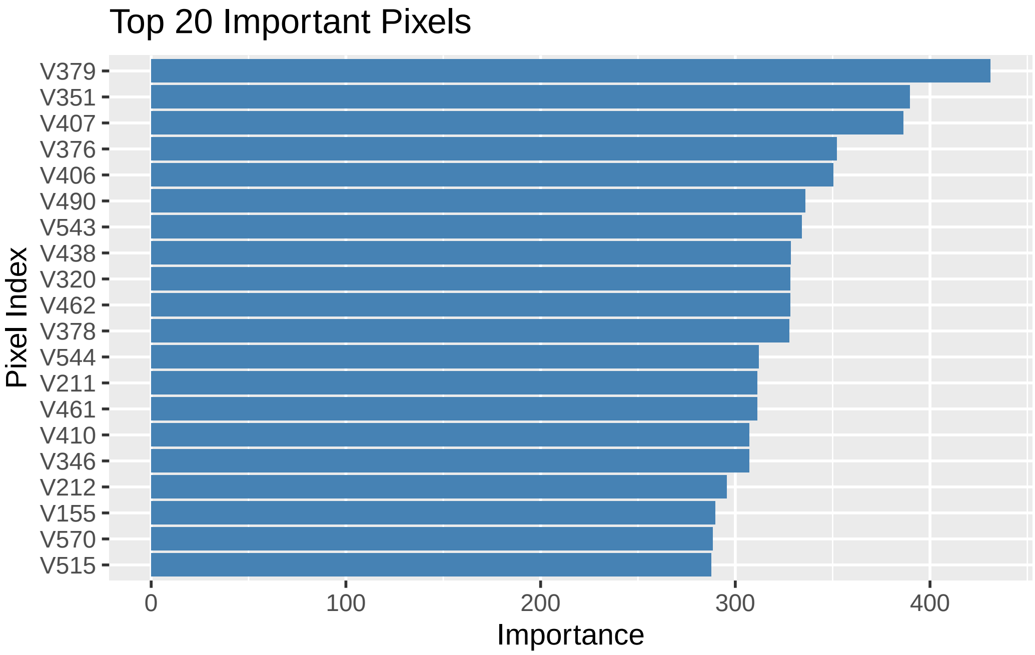
Prediction on Test Set

```
test_df <- as.data.frame(X_test)
digit_predictions <- predict(rf_model, data = test_df)$predictions
digit_predictions <- as.integer(as.character(digit_predictions))

# Save output
saveRDS(digit_predictions, file = "digit_predictions.rds")
```

Variable Importance

```
# Plot top 20 most important pixels
imp_df <- as.data.frame(rf_model$variable.importance)
```

```r
colnames(imp_df) <- "Importance"
imp_df <- imp_df %>%
  rownames_to_column("Pixel") %>%
  arrange(desc(Importance)) %>%
  slice(1:20)


imp_df %>%
  ggplot(aes(x = reorder(Pixel, Importance), y = Importance)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 20 Important Pixels", x = "Pixel Index", y = "Importance")
```



Brief Model Explanation

I used a random forest classifier via the `ranger` package for its speed and ability to capture nonlinear interactions in high-dimensional data like MNIST images. The model achieved high accuracy with fast prediction.

The training labels were converted to factor class, which is required for classification in the random forest model.

The top 20 most important pixels identified by the model are mostly located in central and lower areas of the images, aligning with typical strokes in digits.