# Final Project

Jessica Petrochuk

## Introduction

The iCite Database is a collection of bibliometric information from papers in the PubMed database including MEDLINE, PubMed Central, and CrossRef from 1980 to the present. This includes basic information such as authors, journal, and citation count, but also metadata such as percentile rank of the paper's relative citation ratio (RCR). Due to the size of the dataset, the latest 50,000 papers were used removing any articles that were a withdrawal, correction, corrigendum, or erratum. Of all the attributes provided the ones explored were: if the paper is a research article, if the paper is a clinical paper, the number of authors, the number of references, the average number of other papers in the dataset each author has, and the location of the paper within the Triangle of Biomedicine (animal, human, molecular/cellular).

While, in an ideal world, the contents of a paper should be the most powerful predictor of the number of citations it receives, this study explores the association between a paper's attributes, outside of its contents, and its citation count. Thus the main question posed is: to what extent can the number of citations a paper receives be determined by its attributes, outside of its content.

As seen in Figure 1, the number of citations is count data and thus the first method to address the main goal of this study is to fit a negative binomial regression. This regression was chosen largely due to a large number of papers with 0 citations. The other major methodology used in this study is machine learning, more specifically, the gradient boost algorithm (gbm). Given the many different attributes for this regression problem, a machine learning algorithm can parse through many of these variables and determine which are the most important predictors of citation count as well as potentially capture complex patterns within the data.

Importantly, while doing data exploration a stark difference was noticed between COVID papers and non-COVID papers (where a COVID paper is defined as a paper with either "COVID" or "Coronavirus" in its title). Given that the papers in this particular dataset are all pulled from the year 2022, as expected and as seen in Figure 2, papers categorized as COVID papers had more citations than non-COVID papers on average. Because of this difference, further analysis was done on only non-COVID papers though, in the future, it could be interesting to further investigate the difference between the two categories.

Figure 1: Distribution of Citation Counts

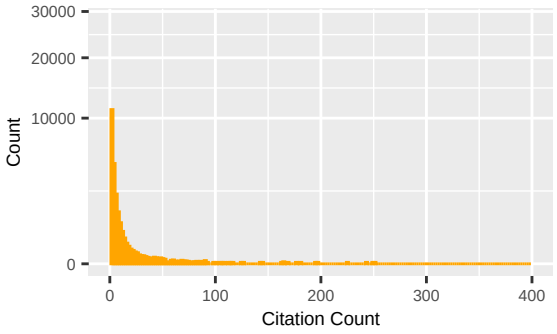
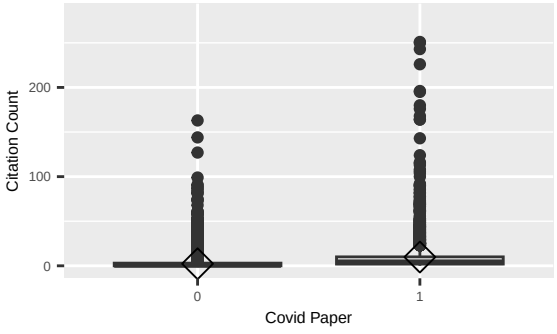



Figure 2: Citations Across COVID vs Not COVID Papers



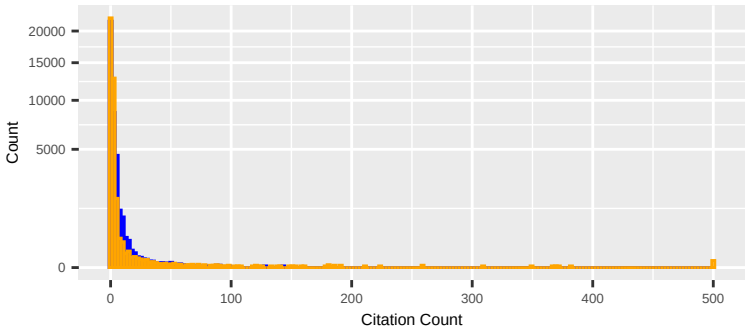
# Results

Before proceeding with analysis the data were preprocessed. As mentioned above, the following analysis was only conducted on papers categorized as non-COVID papers. Furthermore, several variables investigated such as the number of authors and the number of references had to be imputed from provided columns such as lists of authors and lists of references. Following preprocessing, an 80-20 split was used to create a train and test set. The same train and test sets were used for both the regression and machine learning analysis.

## Regression Analysis

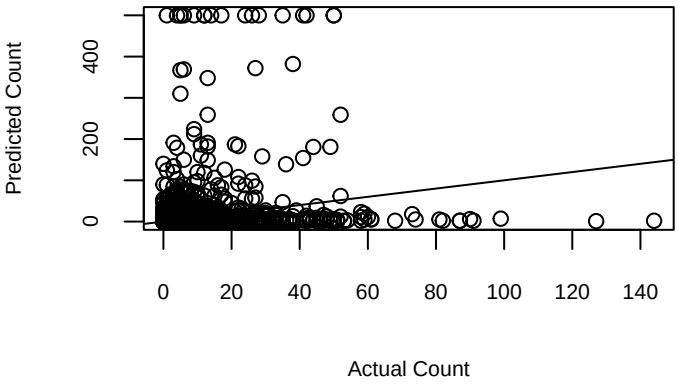
First, a regression analysis was conducted. Knowing the outcome was a count, a negative binomial regression was conducted. After fitting a regression using all of the variables of interest listed above and going through a backward selection process optimizing the R-squared value of the model, the variables of importance were: the number of authors, number of references, and the molecular/cellular extent of the paper. When predicting the number of citations of a paper these variables produced a model with an R-squared value of 0.1487 and RMSE of 4.239 on the test set. It is important to note that, given the negative binomial distribution some values predicted went up to 325382. Given that numbers this high are unreasonable predictions for citations, the predictions were capped at 500. While this R-squared value is low Figure 3 shows that the predicted distribution (orange) approximates the beginning of the actual citation distribution (blue) well.

Figure 3: Actual and Predicted Distributions of Citation Counts

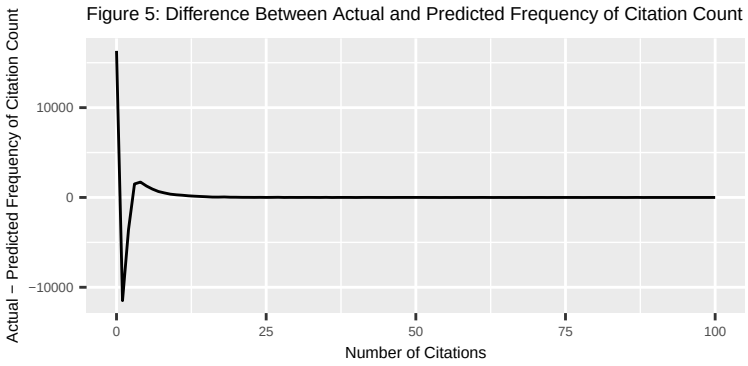



When taking a closer look at the difference between the actual citation count and the predicted citation count in Figure 4, we can see that, in general, the regression predicts a citation count larger than the actual citation count. This can be seen by a larger number of points above the y = x line in the graph.

**Figure 4: Actual Versus Predicted Citation Counts In Train Set**

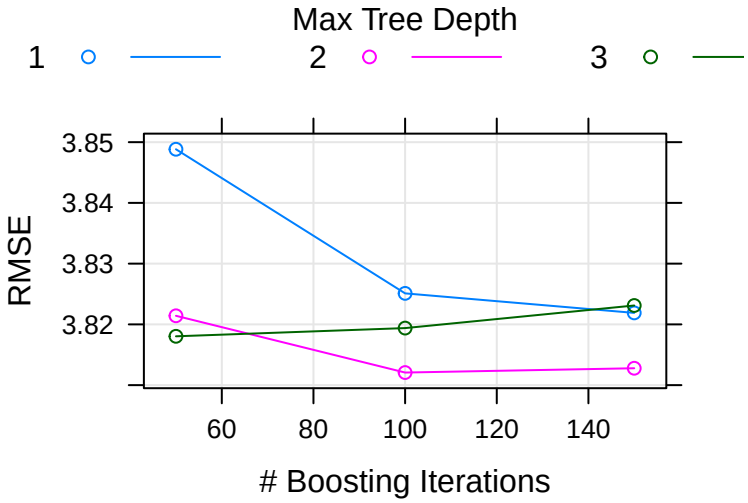



Lastly, by looking at the difference between the frequency of each count in the regression model versus the actual dataset in Figure 5, we can see that in general, the model over predicted the number of papers with 0 citations and underpredicted the number of papers with 1 citation but begins to even out for later citation counts.




## Machine Learning

For the machine learning analysis a gbm was used due to its predictive power and high efficiency. During training the model started with the same training set and 8 variables as the regression analysis. The model was also fitted using cross-validation across 5 groups with a p of 90%. After training, as seen in Figure 7, the optimal model with the lowest RMSE on the training set was with a max tree depth of 2 and 100 trees. This resulted in a model with an R-squared value of 0.1295 on the test set.




Overall, as seen in Table 1, the 3 variables of greatest importance were the same variables that were selected by backward selection in the regression analysis: number of authors, number of references, and the molecular/cellular extent of paper.

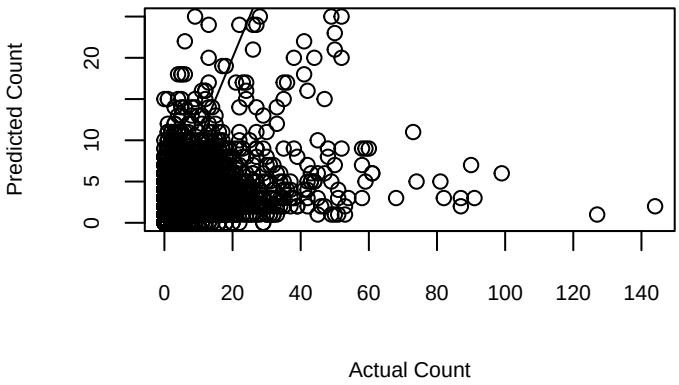
Table 1: Variables of Importance

| | Overall |
|---|---|
| num_authors | 89166.2882 |
| num_ref | 212618.3719 |
| avg_papers_per_author | 9843.3262 |
| is_clinical | 0.0000 |
| is_research_article | 454.6362 |
| human | 563.0566 |
| animal | 2776.2580 |
| molecular_cellular | 20043.7683 |

Lastly, as seen in Figure 9, as opposed to the regression, the gbm model tends to predict fewer citations than

the actual citation count. This can be seen by more points under the y = x line in Figure 9 as in comparison to Figure 4.

**Figure 9: Actual Versus Predicted Citation Counts In Train Set**




# Conclusion

Overall this study looked at the association between paper attributes and citation count. This was done through a negative binomial regression analysis as well as a gbm model. The final negative binomial regression had an R-squared value of 0.1487 and an RMSE of 4.239 on the test set, while the gbm model had an R-squared value of 0.1295 on the test set and an RMSE of 4.130. Within both models, the variables that showed to be most important in predicting citation count were: the number of authors, number of references, and the molecular/cellular extent of paper

On one hand, both the regression and the machine learning model resulted in low R-squared values close to 0 indicating that the attributes did not have strong success in predicting citation count. On the other hand, as shown by the trends derived through the models, the predictive power of these attributes is not null. Interestingly, both models found the same variables of importance including that papers with more authors and more references tend to have more citations overall. Additionally, out of all three parts of the Triangle of Biomedicine, only the cellular/molecular content of a paper showed a strong association with citation count. With more time, it would be interesting to explore further why this might be. Thus the analysis was partially successful, though could be improved had there been more time. One reason these models may have not been as strong is because of the lack of information about the date of the publication. The longer a paper has been published, the more likely it is to have been read by others and cited. Similarly, if a paper was published just this month, even if it has the attributes to have a large number of citations, it simply has not been published for very long. Had there been more time this would also be something to explore further

# References

iCite, Hutchins BI, Santangelo G. iCite Database Snapshots (NIH Open Citation Collection) [Internet]. figshare; 2019. doi: 10.35092/yhjc.c.4586573.

# Appendix (does not count towards page limit, 5 pts)

```
# Run when starting new session
library(gbm)
library(pscl)
library(MASS)
library(caret)
```

```
library(dplyr)
library(knitr)
library(stats)
library(iCiteR)
library(ggplot2)
library(jsonlite)
library(tidyverse)
library(randomForest)

# Set seed for reproducability
set.seed(1213)

# Load previously preprocessed data
papers_filtered = read.csv("filtered_data.csv")
authors = read.csv("authors.csv")
authors_freq = table(authors) |> data.frame()
avg_count_all = read.csv("data_avg.csv")
# Load data from iCiteR
papers = search_metrics(year=2022, limit=50000, page=TRUE)

# Filter papers that are no longer available for citation
papers_filtered = papers |> filter((!title %in% c("Withdrawal.", "Correction.", "Corrigendum.", "Erratu

# Fixing columns to correct data type
papers_filtered = papers_filtered |>
  mutate(is_research_article = case_when(
    is_research_article == "No" ~ FALSE,
    is_research_article == "Yes" ~ TRUE
)) |>
  mutate(is_clinical = case_when(
    is_clinical == "No" ~ FALSE,
    is_clinical == "Yes" ~ TRUE
)) |>
  mutate(provisional = case_when(
    provisional == "No" ~ FALSE,
    provisional == "Yes" ~ TRUE
))

# Adding additional potential columns of interest
papers_filtered = papers_filtered |>
  rowwise() |>
  mutate(num_authors = length(c(strsplit(authors, ", ")[[1]])),
         num_ref = length(c(strsplit(references, " ")[[1]])),
         num_cit_clin = length(c(strsplit(cited_by_clin, " ")[[1]])),
         covid_paper = grepl("COVID|Coronavirus", title) * 1
  )

# Factorize journal column
papers_filtered$journal = as.factor(papers_filtered$journal)

# Save data frame to not load every time
write.csv(papers_filtered, file="filtered_data.csv", row.names=FALSE)
authors = c()
```

```
avg_paper_count = c()

# Get all authors
for (author_list in papers_filtered$authors) {
  author_list_curr = c(strsplit(author_list, ", ")[[1]])
  authors = append(authors, author_list_curr)
}

# Save data frame to not load every time
write.csv(authors, file="authors.csv", row.names=FALSE)
avg_count_all = c()

# Get average number of papers each author has in dataset
for (author_list in papers_filtered$authors) {
  author_list_curr = c(strsplit(author_list, ", ")[[1]])
  total_authors = length(author_list_curr)
  total_count = 0
  for (author in author_list_curr) {
    count = as.integer(filter(authors_freq, x == author)$Freq)
    total_count = total_count + count
  }
  average = total_count / total_authors
  avg_count_all = append(avg_count_all, average)
  print(length(avg_count_all))
}

papers_filtered$avg_papers_per_author = avg_count_all
write.csv(avg_count_all, file="data_avg.csv", row.names=FALSE)
# Histogram of Citations
ggplot(aes(x = citation_count), dat=papers_filtered) +
  geom_histogram(bins=200, fill="orange", color="orange", alpha=0.25) +
  scale_y_sqrt() +
  xlab("Citation Count") +
  ylab("Count") +
  ggtitle("Figure 1: Distribution of Citation Counts") +
  xlim(0, 400) +
  theme(text = element_text(size = 7))

covid = filter(papers_filtered, covid_paper==TRUE)
not_covid = filter(papers_filtered, covid_paper==FALSE)
papers_filtered$covid_paper = as.factor(papers_filtered$covid_paper)

ggplot(aes(x = covid_paper, y=citation_count), dat=papers_filtered) +
  geom_boxplot() +
  ylim(0, 280) +
  ylab("Citation Count") +
  xlab("Covid Paper") +
  ggtitle("Figure 2: Citations Across COVID vs Not COVID Papers") +
  theme(text = element_text(size = 6)) +
  stat_summary(fun.y=mean, geom="point", shape=23, size=4)
papers_full = dplyr::select(not_covid, c(citation_count, num_authors, num_ref, avg_papers_per_author, is

# Columns of interest selected by backward selection
```

```
papers_neg_bin = dplyr::select(papers_full, c(citation_count, num_authors, num_ref, molecular_cellular))

# Extract x and y from full dataset
x = dplyr::select(papers_full, -c(citation_count))
y = papers_full$citation_count

# Create train test split
set.seed(1213)
test_index = createDataPartition(y, times = 1, p = 0.2, list = FALSE)

# Split x and y into train and test
y_train = y[-test_index]
y_test = y[test_index]
x_train = x[-test_index, ]
x_test = x[test_index, ]

# Split neg bin data into train and test
x_neg_bin = dplyr::select(papers_neg_bin, c(num_authors, molecular_cellular, num_ref))
x_train_neg_bin = x_neg_bin[-test_index, ]
x_test_neg_bin = x_neg_bin[test_index, ]
all_train_neg_bin = papers_neg_bin[-test_index, ]
all_test_neg_bin = papers_neg_bin[test_index,]
# Negative binomial model
neg_bin_full = glm.nb(citation_count ~ ., data=all_train_neg_bin)
#summary(neg_bin_full)
predictions_neg_bin = as_tibble(as.integer(neg_bin_full$fitted.values))
y_hat_neg_bin = as.tibble(predict(neg_bin_full, x_test_neg_bin))
test = filter(predictions_neg_bin, value > 500)

# Capping predictions at 500 citations
predictions_neg_bin = predictions_neg_bin |> mutate(value = case_when(
  value > 500 ~ 500,
  TRUE ~ as.numeric(value)
))

y_hat_neg_bin = y_hat_neg_bin |> mutate(value = case_when(
  value > 500 ~ 500,
  TRUE ~ as.numeric(value)
))

# Figure 3: Histogram of Citations
ggplot(aes(x = citation_count), dat=all_train_neg_bin) +
  geom_histogram(bins=200, fill="blue", color="blue", alpha=0.8) +
  geom_histogram(aes(x=value), dat=predictions_neg_bin, bins=200, fill="orange", color="orange", alpha=
  scale_y_sqrt() +
  xlab("Citation Count") +
  ylab("Count") +
  ggtitle("Figure 3: Actual and Predicted Distributions of Citation Counts") +
  theme(text = element_text(size = 6))
# Figure 4: Actual Values vs Predicted Values
# postResample(y_hat_neg_bin$value, y_test)
plot(y_train,
     predictions_neg_bin$value,
```

```
     cex.main=0.7,
     cex.lab = 0.7,
     cex.axis = 0.7,
     xlab="Actual Count",
     ylab="Predicted Count",
     main="Figure 4: Actual Versus Predicted Citation Counts In Train Set")
abline(0, 1)
# Figure 6: Difference between actual and predicted
differences = c()
for (citation_count in c(0:100)) {
  curr_difference = sum(y == citation_count) - sum(predictions_neg_bin$value == citation_count)
  differences = append(differences, curr_difference)
}
difference_table = tibble(x = c(0:100), diff = differences)

ggplot(aes(x=x, y=diff), dat = difference_table) +
  geom_line() +
  xlab("Number of Citations") +
  ylab("Actual - Predicted Frequency of Citation Count") +
  ggtitle("Figure 5: Difference Between Actual and Predicted Frequency of Citation Count") +
  theme(text = element_text(size = 6))
x_train$is_clinical = as.integer(x_train$is_clinical)
x_train$is_research_article = as.integer(x_train$is_research_article)

control = trainControl(method = "cv", number = 5, p = .9)
train_gbm =  train(x_train, y_train,
                   method = "gbm",
                   trControl = control
                   )

saveRDS(train_gbm, "gbm.rds")


train_gbm = readRDS("gbm.rds")

# Figure 7: Number of Trees
plot(train_gbm,
     cex.main=0.2,
     cex.lab = 0.2,
     cex.axis = 0.2,
     ylab="RMSE",
     )
# Figure 8: Looking at variables of importance for gbm model
gbmimp = varImp(train_gbm, scale=FALSE)
kable(gbmimp$importance, caption="Variables of Importance")

#y_hat_gbm = as.integer(predict(train_gbm, x_test))
#postResample(y_hat_gbm, y_test)
y_hat_gbm_train = as.integer(predict(train_gbm, x_train))
plot(y_train,
     y_hat_gbm_train,
     cex.main=0.7,
     cex.lab = 0.7,
```

```
     cex.axis = 0.7,
     xlab="Actual Count",
     ylab="Predicted Count",
     main="Figure 9: Actual Versus Predicted Citation Counts In Train Set")
abline(0, 1)

#plot(y_test, y_hat)
#abline(0, 1)
#title("Figure 9: Actual Versus Predicted Citation Counts")
# Zero inflated poisson model
zero_inf_full = zeroinfl(citation_count ~ ., data=all_train_neg_bin, dist="poisson")
summary(zero_inf_full)
predictions = as_tibble(as.integer(zero_inf_full$fitted.values))
y_hat_zero_inf = predict(zero_inf_full, x_test_neg_bin)
postResample(as.integer(y_hat_zero_inf), y_test)
plot(y_test, y_hat_zero_inf, )
abline(0, 1)


grid = data.frame(mtry = c(1, 5, 10, 25))

train_rf =  train(x_train[0:5000,], y_train[0:5000],
                  method = "rf",
                  ntree = 150,
                  trControl = control,
                  tuneGrid = grid,
                  nSamp = 500) # Change back to larger size was 5000

plot(train_rf)

y_hat = as.integer(predict(train_rf, x_test))
postResample(y_hat, y_test)


rfimp = varImp(train_rf, scale=FALSE)
rfimp

saveRDS(train_rf, "random_forest.rds")
train_rf = readRDS("random_forest.rds")
```

- All figures/tables and code
- See https://bookdown.org/yihui/rmarkdown-cookbook/code-appendix.html

## GitHub webpage (10 pts)

- See: https://pages.github.com/ and free templates here https://jekyllthemes.io/free
- Display key points from Intro, Results, Conclusion, and References
- Include key figures/tables
- Can use same text as your report

```
# Zero inflated poisson model
zero_inf_full = zeroinfl(citation_count ~ ., data=all_train_neg_bin, dist="poisson")
summary(zero_inf_full)
predictions = as_tibble(as.integer(zero_inf_full$fitted.values))
```

```
y_hat_zero_inf = predict(zero_inf_full, x_test_neg_bin)
postResample(as.integer(y_hat_zero_inf), y_test)
plot(y_test, y_hat_zero_inf, )
abline(0, 1)


grid = data.frame(mtry = c(1, 5, 10, 25))

train_rf =  train(x_train[0:5000,], y_train[0:5000],
                  method = "rf",
                  ntree = 150,
                  trControl = control,
                  tuneGrid = grid,
                  nSamp = 500) # Change back to larger size was 5000

plot(train_rf)

y_hat = as.integer(predict(train_rf, x_test))
postResample(y_hat, y_test)


rfimp = varImp(train_rf, scale=FALSE)
rfimp

saveRDS(train_rf, "random_forest.rds")
train_rf = readRDS("random_forest.rds")
```