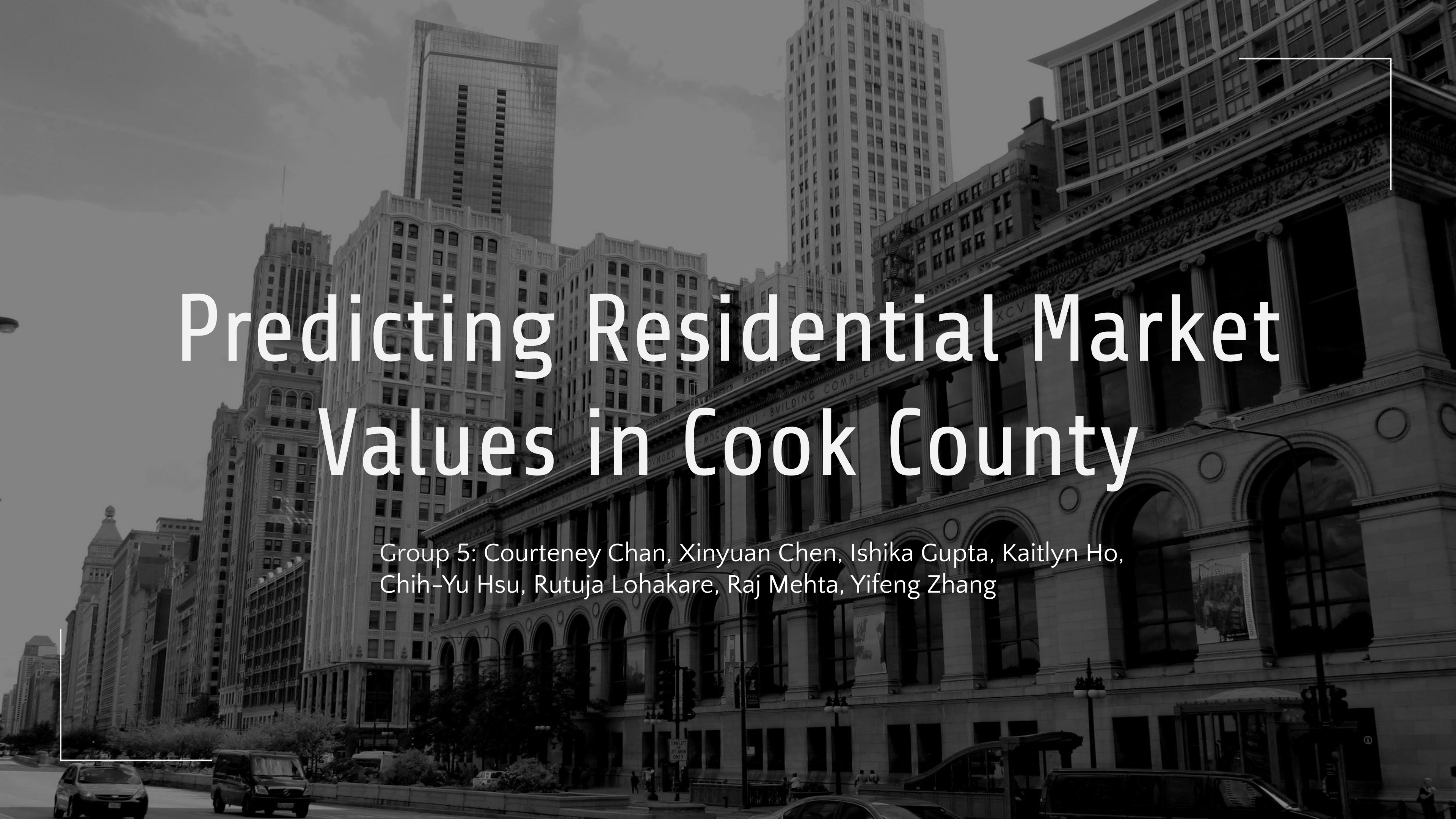


Predicting Residential Market Values in Cook County

Group 5: Courteney Chan, Xinyuan Chen, Ishika Gupta, Kaitlyn Ho, Chih-Yu Hsu, Rutuja Lohakare, Raj Mehta, Yifeng Zhang

Table of contents

01 Business overview

Cook County Assessor's Office Timeline

Source

James Houlihan (1997-2010)

1.1% of his reassessments were identical over 2 assessment periods

Joseph Berrios (2010-2018)

67.4% of his reassessments were identical over 2 assessment periods

Fritz Kaegi (2018)

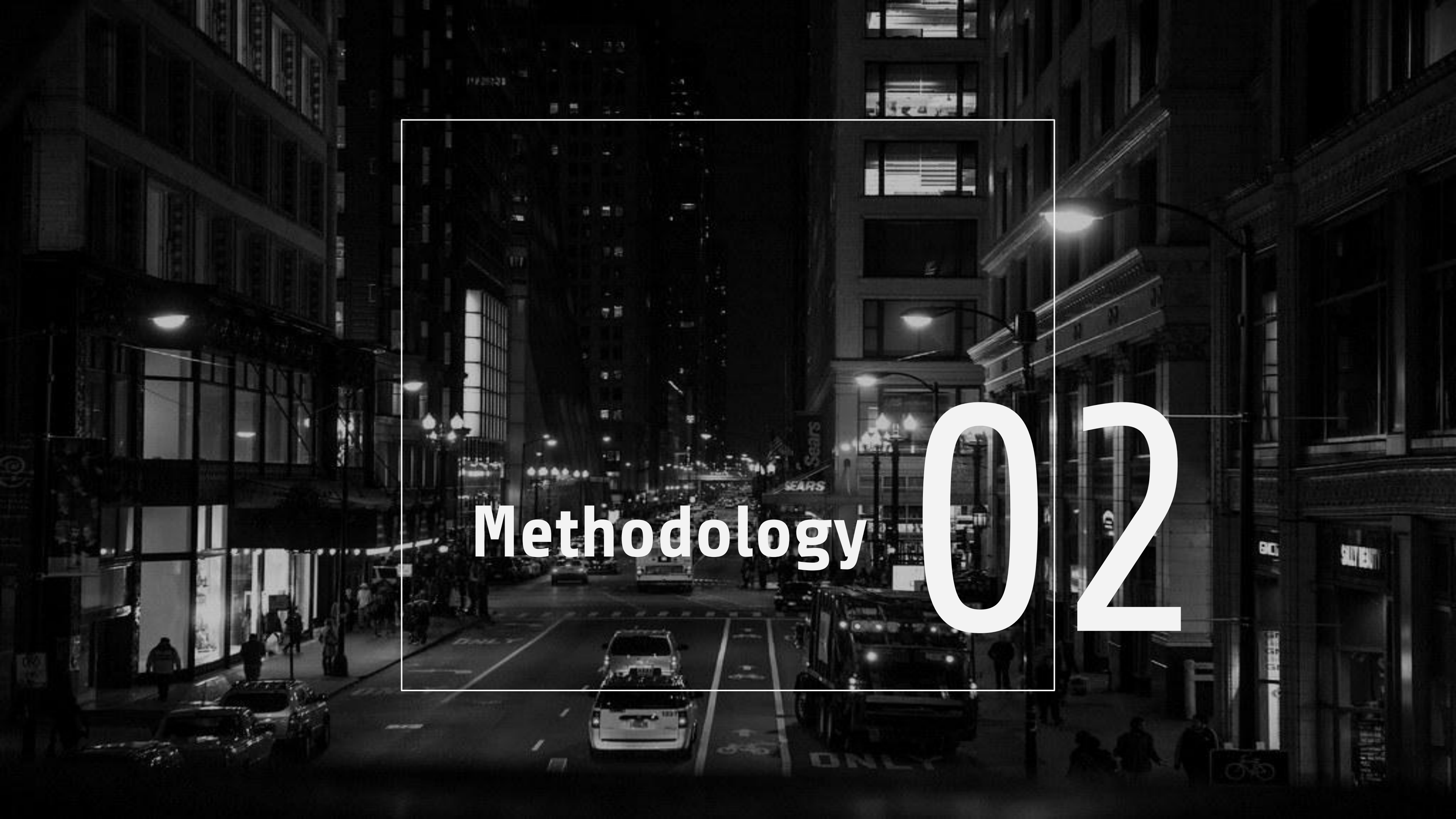
Beats Berrios in election for Cook County Assessor due to ethics ruling cases

Problem

Current system creates inaccurate assessed values for housing units, disportionately affecting homeowners in lower socioeconomic brackets.

Objectives

- Create more accurate property taxes by improving the computer program used to determine a property’s assessed value
    - Develop an algorithm that can predict the fair market value of
    - Identify the characteristics that will increase the accuracy of the predictions



Methodology 02

Preprocessing

- Eliminate columns not indicated as predictors in the codebook

- Eliminate duplicate rows

- Missing values
  - Impute by meta_nbhd & meta_town_code (Last Observation Carried Forward)

  - Impute numerical values with the mean of the column

Preprocessing Continued

- Outliers
  - Winsorize the values below 1% and above 99%

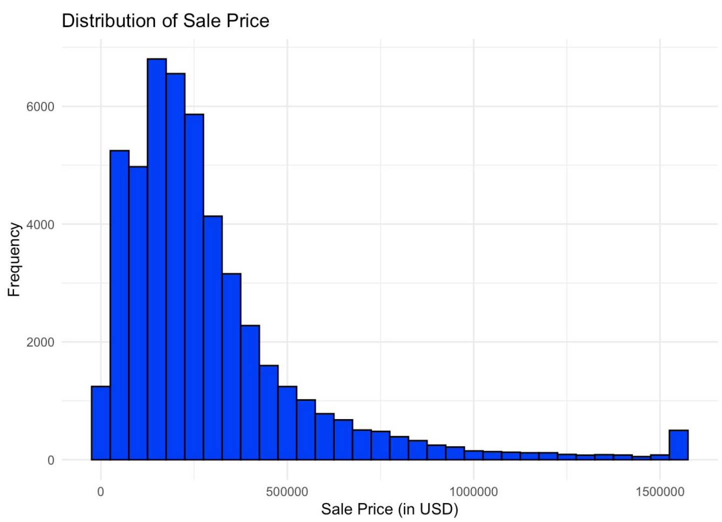
```
historic_property_data[numeric_cols] <- lapply(historic_property_data[numeric_cols],
                                               function(x) Winsorize(x, probs = c(0.01, 0.99)))
```

- Negative Values
  - Change to 0

- Removed remaining columns with too many missing values: "char_apts", "char_porch", "char_attic_fnsh", "char_tp_dsgn"

- Results: Dataframe with 49,362 rows and 38 columns

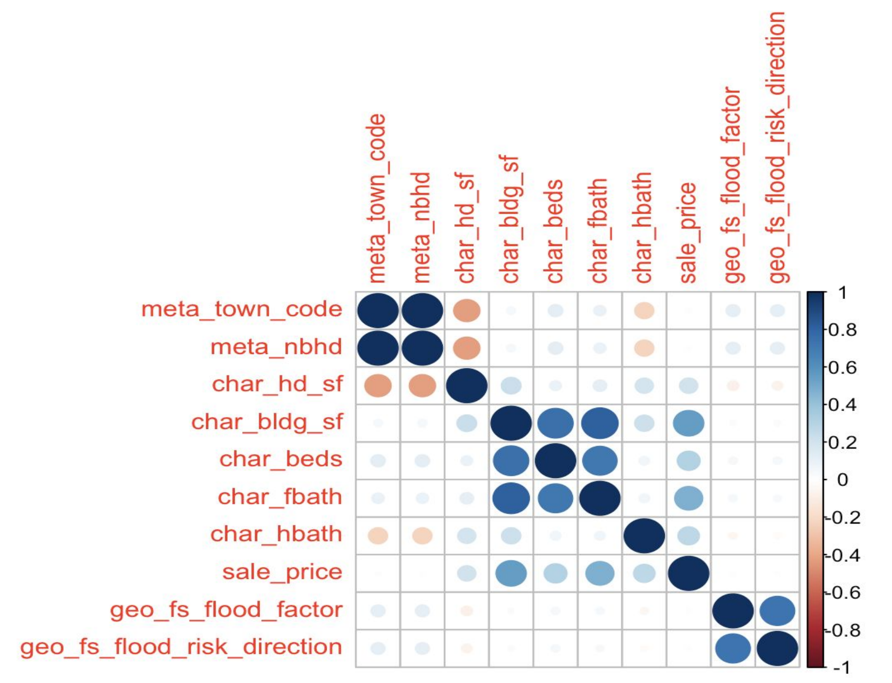
Histogram of Sales Price



Hypothesis: Our predicted assessment values should have a similar distribution

Selecting Predictors

- Location, size, and potential risk to construction quality
  - 'Meta_nbhd', 'geo_school_elem_district', 'geo_school_hs_district'
  - 'char_hd_sf', 'char_bldg_sf', 'char_beds', 'char_fbath', 'char_hbath'
  - 'geo_fs_flood_factor','geo_fs_flood_risk_direction'



Correlation Matrix of Numerical Variables

Testing for Multicollinearity

Variance Inflation Factor(VIF)→ Remove township code

meta_town_code	meta_nbhd
66205.936494	66171.245959
char_hd_sf	char_bldg_sf
1.355389	4.253432
char_beds	char_fbath
2.635338	3.285497
char_hbath	geo_fs_flood_factor
1.167887	2.160514
geo_fs_flood_risk_direction	
2.153850	

Drop ‘meta_town_code’

meta_nbhd	char_hd_sf
1.311712	1.346188
char_bldg_sf	char_beds
4.246855	2.629873
char_fbath	char_hbath
3.280663	1.167812
geo_fs_flood_factor	geo_fs_flood_risk_direction
2.156316	2.153380

03 Key Observations

Predictive Test Models

- Set seed and prepare data for cross-validation

```
set.seed(123) # Set a seed for reproducibility

# Split the data into training and testing sets
index <- createDataPartition(data_selected_reduced$sale_price, p = 0.8, list = FALSE)
train_data <- data_selected_reduced[index, ]
test_data <- data_selected_reduced[-index, ]
```

```
# Set up cross-validation
train_control <- trainControl(method = "cv", number = 5, allowParallel = TRUE)
```

- Linear, Random Forest, and Lasso Regression Models (see next slide)

Predictive Models

Linear Regression	Random Forest
○ MSE: 43,220,479,448 ○ RMSE: 207,895.4	○ MSE: 15,276,646,289 ○ RMSE: 123,598.7
Lambda	**Combined**
○ MSE: 43,217,434,819 ○ RMSE: 207,888	○ MSE: 28,847,341,624 ○ RMSE: 169,844.8

Final Model Design

```r
new_data <- predict_property_data

new_data_cols <- c('meta_nbhd', 'char_hd_sf', 'char_bldg_sf', 'char_beds', 'char_fbath', 'char_hbath','geo_fs_flo
od_factor','geo_fs_flood_risk_direction')

new_data_prepared <- select(new_data, all_of(new_data_cols))
# Predicting values using the best model or combined approach
final_predictions <- (predict(lm_model, new_data_prepared) +
                      predict(rf_model, new_data_prepared) +
                      predict(lasso_model, new_data_prepared)) / 3

# Combine final predictions with the pid column from new data
assessed_values <- data.frame(pid = new_data$pid, assessed_value = final_predictions)

# Ensure that all pid and assessed_value values are non-missing and non-negative
assessed_values <- assessed_values %>%
  filter(!is.na(assessed_value) & assessed_value >= 0)
```

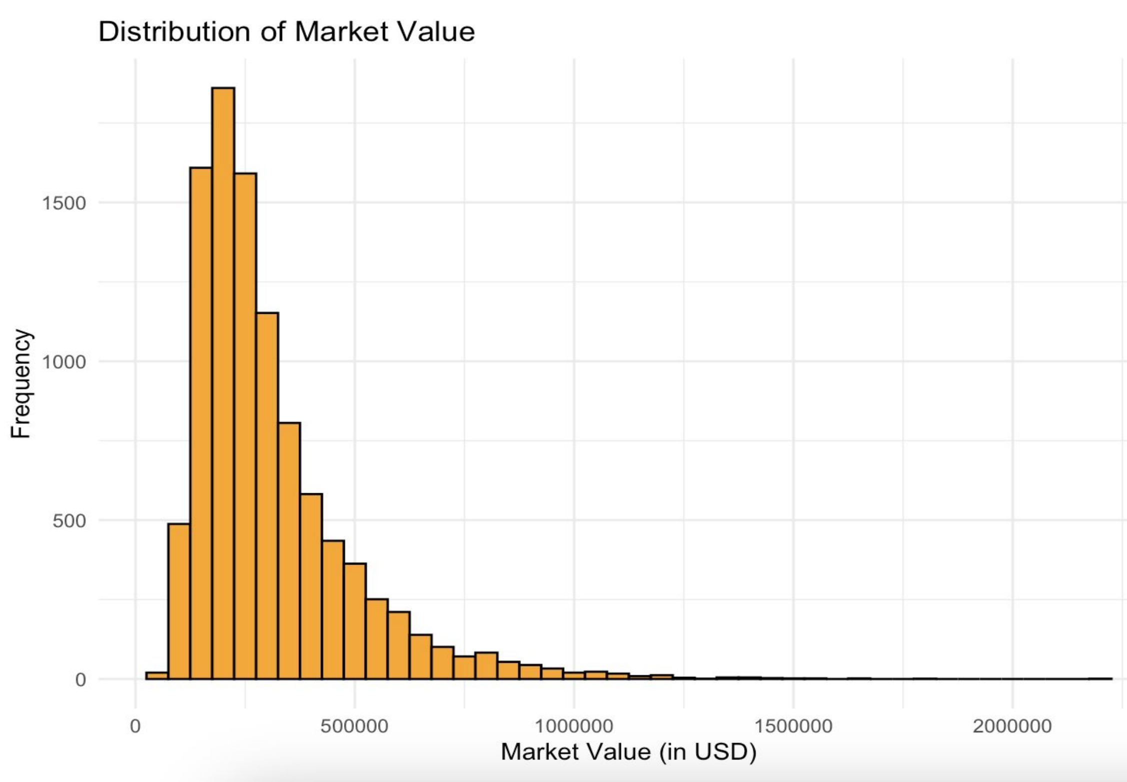
Assessment Values

Summary Statistics:

- Min: 42,012
- 1st Quartile: 184,985
- Median: 255, 128
- Mean: 308,566
- 3rd Quartile: 372,765
- Max: 2,180,952

pid	assessed_value
1	247432.935
2	339336.788
3	161193.162
4	275215.696
5	424739.132
6	194805.689
7	246633.161
8	425016.315
9	318902.218
10	565756.817
⋮	⋮
9990	175777.32719738374
9991	172559.34466199155
9992	386877.4458173979
9993	190178.86938117535
9994	179101.76594420907
9995	523595.8526599297
9996	509194.16243215586
9997	146496.22663066807
9998	397966.7463866722
9999	581761.7615871712
10000	230271.74869759157

Distribution of Assessed Values




Thank you.

Sources

- Slide 1 Photo
- Slide 2 Photo
- Slide 3 Photo (Redfin)
- Slide 4 Photo 1, 2, 3 (Information)
- Slide 6 Photo (Flickr) (section 2)
- Slide 10 Photo (Flickr) (section 3)
- Slide 18 Photo (Flickr)

Work Distribution

- Courteney Chan – Data Preprocessing
- Xinyuan Chen – Selecting Predictors, Algorithms
- Ishika Gupta – Selecting Predictors, Algorithms
- Kaitlyn Ho – Selecting Predictors, Visualizations, executive summary, slides
- Chih-Yu Hsu – Data preprocessing
- Rutuja Lohakare – Data Preprocessing, Selecting Predictors, Algorithms
- Raj Mehta – Selecting Predictors, Algorithms
- Yifeng Zhang – Executive Summary