Background:

The *ASHRAE Great Energy Predictor III* competition revolves around developing accurate predictions of metered building energy usage in the following areas: electric, chilled water, hot water, and steam meters. These predictions will be built using variables like the year a building was built, the size of the building (square footage), meteorological data, the number of floors of the building, and primary use. With better estimates of energy-saving investments, investors and institutions will be more inclined to invest in eco-friendly features and upgrades.

The sponsor of the Kaggle competition is the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), an American professional association pursuing the advancement of heating, ventilation, air conditioning and refrigeration. A core component to ASHRAE's purpose pertains to the development and publishing of technical standards to improve building services engineering, energy efficiency, indoor air quality, and sustainable development.

Dataset:

ASHRAE's Great Energy Predictor included five separate data sources from over 1,000 buildings over a three-year timeframe. Fortunately for us, two of these datasets pertained to the test datasets in which ASHRAE would measure performance. We discarded the testing datasets altogether since they did not include meter readings, the variable we are interested in predicting.

Discarding the testing datasets left us with three data sources. The first data source contained information regarding the individual buildings metadata. Buildings metadata contains a site identifier, building identifier, the primary use of the building (education, office, retail, etc.), square footage, year built, and number of floors. The second data source included information such as the building identifier, meter type (electricity, chilled water, steam, hot water), a timestamp, and energy consumption. The final data source contained weather data from nearby meteorological station such as site identifier, air temperature, cloud coverage, dew temperature, precipitation depth, sea level pressure, wind direction, and wind speed.

Existing Notebook Review

A Deep Dive EDA Into All Variables by Jason Zivkovic

Prior to conducting exploratory data analysis (EDA), the train, building metadata, and weather datasets were joined to create a data frame with 20,216,100 rows and 22 columns.

Missing Data

Missing data was an unavoidable issue with the provided datasets, specifically with missing building and weather variables. This issue will be addressed later but should not be overlooked.

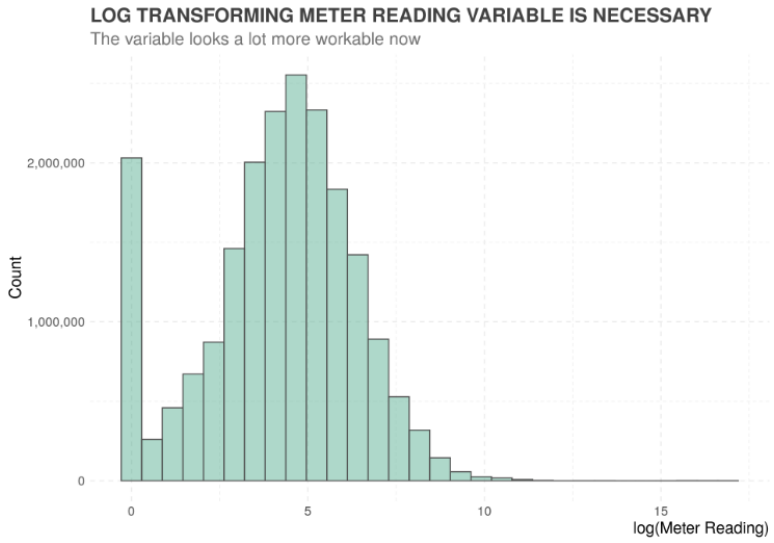
```
colSums(is.na(combine_train))
```

```
##          building_id                meter            timestamp
##                    0                    0                    0
##        meter_reading              site_id          primary_use
##                    0                    0                    0
##          square_feet           year_built          floor_count
##                    0             12127645             16709167
##      air_temperature       cloud_coverage      dew_temperature
##                96658              8825365               100140
##    precip_depth_1_hr   sea_level_pressure       wind_direction
##              3749023              1231669              1449048
##           wind_speed       timestamp_date      timestamp_month
##               143676                    0                    0
##        timestamp_day timestamp_day_number         time_ymd_hms
##                    0                    0                    0
##            time_hour
##                    0
```

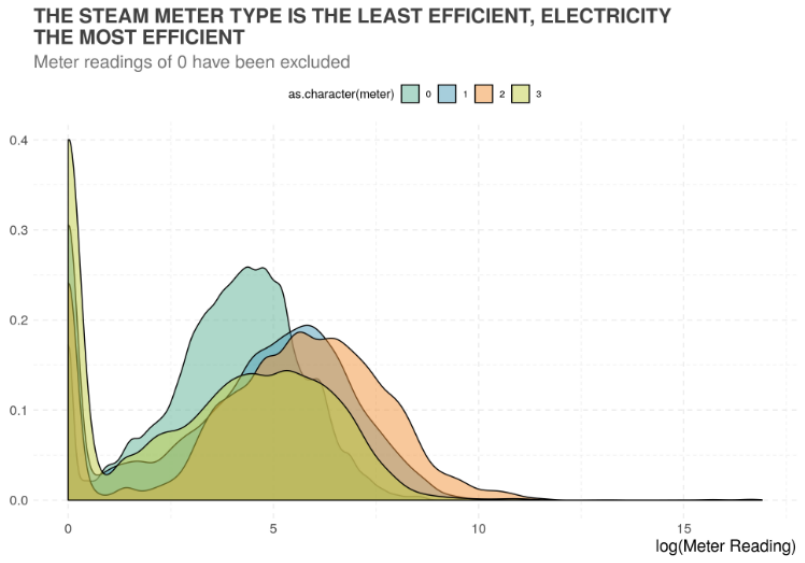
Variable Analysis

Zivkovic exploration of each variable began with visualizations of said variables distribution. Meter reading is the most noteworthy of all variables explored, it was heavily skewed by several outliers. The code and output of meter reading's histogram is below. Since there was no way Zivkovic could investigate these outliers he ultimately eliminated all observations greater than 2 interquartile ranges above the mean. However, after doing so the meter reading variable was still heavily skewed and ended up requiring a logarithmic transformation.

```
combine_train %>%
  ggplot(aes(x= log(meter_reading + 1))) +
  geom_histogram(alpha = 0.5, fill = "#5EB296", colour = "#4D4D4D") +
  scale_y_continuous(labels = comma) +
  ggtitle("LOG TRANSFORMING METER READING VARIABLE IS NECESSARY", subtitle = "The variable looks a lot more workable now") +
  labs(x= "log(Meter Reading)", y= "Count")
```



Zivkovic utilized several other variables to produce some interesting visualizations. The first visualization I'd like to talk about is meter type and meter reading. Producing this plot showed a distinct correlation between meter type and the amount of energy consumed, steam meter type tends to have higher readings while electricity tends to have the lowest. Zivkovic also created visualizations showing the relationship between energy usage and day, time of day and energy usage, and several other histograms using the buildings primary use, square footage, the year each building was built.



Takeaway

While this notebook didn't go into building and tweaking a predictive model, it did provide some great insights into issues that we should be mindful of when we start our machine learning process.

Critique

My biggest critique with this notebook is the overall lack of data cleaning/prep before the EDA process and the abandoning of all N/A values within the dataset. Eliminating N/As removed almost 1/3rd of the total observations within the combined data frame.

https://www.kaggle.com/jaseziv83/a-deep-dive-eda-into-all-variables

AC/DC by KXX

Definition of Functions

The author of this notebook chose to create functions used for data preparation, as follows:

```
#----------------------------
cat("Defining functions...\n")
prep <- function(dt_fn, meta_fn, weather_fn, drops, sort_timestamp = FALSE, n_max = Inf){

  cat("Loading data...\n")
  dt <- fread(dt_fn, nrows = n_max, verbose = FALSE)
  weather <- fread(weather_fn, nrows = n_max, verbose = FALSE)
  meta <- fread(meta_fn, nrows = n_max, verbose = FALSE)

  cat("Merging datasets...\n")
  dt[meta, on = "building_id", names(meta) := mget(names(meta))]
  dt[weather, on = c("site_id", "timestamp"), names(weather) := mget(names(weather))]

  rm(meta, weather)
  invisible(gc())

  cat("Processing features...\n")
  dt[, (drops) := NULL
     ][, timestamp := fastPOSIXct(timestamp)
       ][, `:=`(wday = wday(timestamp),
                hour = hour(timestamp),
                year_built = year_built - 1900,
                square_feet = log1p(square_feet))]

  cat("Converting categorical columns...\n")
  dt[, names(dt) := lapply(.SD, function(x) {if (is.character(x)) x <- as.integer(as.factor(x)); x})]

  if (sort_timestamp) setorder(dt, timestamp)
  dt[, timestamp := NULL]
}
```

This is a valuable feature of the analysis because it allows for easy reproducibility of experimental models. For instance, if a user intended to load certain subsets of the data, instead of having to modify a hard-coded block of code, a user could use functions like this one to quickly create and prepare multiple datasets for use in models.

Importantly, the timestamps within the original dataset are split into multiple columns – one indicating weekday and another indicating hour. Year built is transformed into a smaller number by subtracting a constant, while square feet is log transformed to provide more workable data.

Preparing the Data

Using the function defined above, the author then prepared the data for training as follows:

```
cat("Preparing train data...\n")
cats <- c("building_id", "site_id", "meter", "primary_use", "hour", "wday")
drops <- c("sea_level_pressure", "wind_direction", "wind_speed")
tr <- prep("../input/ashrae-energy-prediction/train.csv",
           "../input/ashrae-energy-prediction/building_metadata.csv",
           "../input/ashrae-energy-prediction/weather_train.csv",
           drops, TRUE)
y <- log1p(tr$meter_reading)
tr[, meter_reading := NULL]
tr <- data.matrix(tr)
```

Note the ease with which data preparation is completed using a pre-defined function. The author elected to drop sea level pressure, wind direction, and wind speed from the data, and notes which variables are categorical which is an important factor when building a model (models cannot interpret text!).

Training the Model

Once the data is loaded and prepared, the model is trained as follows:

```
cat("Training model...\n")

p <- list(boosting = "gbdt",
          objective = "regression_l2",
          metric = "rmse",
          nthread = 4,
          learning_rate = 0.05,
          num_leaves = 40,
          colsample_bytree = 0.85,
          lambda = 2)

N <- nrow(tr)
tsf <- caret::createTimeSlices(y, N/2, N/2)
models <- list()
imp <- data.table()

for (i in seq_along(tsf)){
  cat("\nFold:", i, "\n")
  idx <- tsf[[i]][[1]]

  xtrain <- lgb.Dataset(tr[-idx, ], label = y[-idx], categorical_feature = cats)
  xval <- lgb.Dataset(tr[idx, ], label = y[idx], categorical_feature = cats)
  models[[i]] <- lgb.train(p, xtrain, 4000, list(val = xval), eval_freq = 200, early_stopping_rounds = 200)

  imp <- rbind(imp, lgb.importance(models[[i]]))

  rm(xtrain, xval)
  invisible(gc())
}

lgb.plot.importance(imp[, lapply(.SD, mean), by=Feature], ncol(tr), cex=1)

rm(tr, y, imp, p, tsf, N, i, idx)
invisible(gc())
```

The author elected to use a LightGBM (Gradient Boosting Machine) model for this analysis, which essentially builds several different models, combining a few methods, to iteratively create trees with the goal of improving prediction accuracy with each tree. Importantly, the author elects to use cross validation to select a best model. Then, the variable importance found by the model is output.

Testing

Once the model is trained, the author uses it to make predictions on the test set as follows:

```
cat("Preparing test data...\n")
te <- prep("../input/ashrae-energy-prediction/test.csv",
           "../input/ashrae-energy-prediction/building_metadata.csv",
           "../input/ashrae-energy-prediction/weather_test.csv",
           drops)
invisible(gc())
te <- as.matrix(te[, row_id := NULL])
invisible(gc())

#----------------------------
cat("Making predictions...\n")
pred_te <- sapply(models, function(m_lgb) expm1(predict(m_lgb, te)))
invisible(gc())

sub <- fread("../input/ashrae-energy-prediction/sample_submission.csv")
sub[, meter_reading := round(rowMeans(pred_te), 2)
    ][meter_reading < 0, meter_reading := 0]

fwrite(sub, "submission.csv")
```

Takeaways

It will be important for us to take note of the preprocessing steps taken in this and other notebooks. Time and date often give people a lot of difficulty, but this code shows an easy way to handle them. Certain predictors are on very different scales than others. It will be important to scale or transform our data before proceeding with our analysis. Cross-validation/parameter tuning will be an important step to take to ensure that we use the best model to make predictions moving forward.

Critiques

This notebook is not a notebook at all – it simply contains code. I wish the notebook had commentary because it would have made it much easier to understand the aims and functions of the code itself. Additionally, unlike other notebooks, this notebook does not perform any EDA, and so, if I was new to this problem, I would have no real idea what kind of data the author is working with, how they are correlated, or how they are distributed. EDA will be an important step to ensuring the audience understands the data used to build the model.

https://www.kaggle.com/kailex/ac-dc