Final Project - Machine Learning

Omry Yoresh

June 2023

1. Introduction:

This project aims to replicate and extend the main results of my paper "Heads Up: Does Pollution Cause Construction Accidents?" joint with Victor Lavy and Genia Rachkovsky, within the framework of the course "Machine Learning for Economists" at The Hebrew University of Jerusalem. The original study investigates the relationship between pollution and construction accidents, focusing on the potential impact of Nitrogen Dioxide (NO_2) on workplace safety within the construction industry.

Building upon the original research, this project utilizes machine learning techniques and methodologies learned in the course to further explore and validate the findings. Firstly, a double lasso approach will be employed to examine and potentially validate NO_2 as the main pollutant used. Given the availability of multiple pollutants and weather variables, this analysis will help identify the most influential factors associated with construction accidents.

In addition to the original study's exploration of a non-linear model based on EPA's NO_2 critical levels, this project introduces the utilization of the Random Forest method. This approach aims to validate and potentially identify different critical NO_2 cut-off pollution levels that play a significant role in predicting construction accidents. By leveraging the Random Forest method, the project aims to provide additional insights into the relationship between pollution levels and accident occurrence.

Furthermore, this project extends beyond the replication and validation of the original findings by incorporating a Ridge prediction model. The prediction model will assess the likelihood of a construction accident at a given construction site. A comparative analysis will be conducted to evaluate the accuracy of the model based on same-day data versus lagged data, considering the potential policy implications. Understanding the predictive power of different time-frames will potentially enable policymakers to make informed decisions regarding safety protocols and resource allocation.

The project will progress as follows: Chapter 2 focuses on data replication, ensuring consistency with the original study. Chapter 3 employs the double lasso method to validate the chosen pollution variables and explore potential regressors. In Chapter 4, the analysis incorporates the Random Forest method to investigate non-linear relationships and identify critical pollution thresholds. Lastly, Chapter 5 compares and creates prediction models to assess the likelihood of construction accidents using same-day and lagged data. Chapter 6 will conclude the project by summarizing the key findings and discussing their implications.

2. Data Replication:

2.1 Data Preparation:

The data-set used in this paper is a combination of data from three primary sources: the Israeli Ministry of Economy and Industry, which provided the construction sites' locations, activity dates, and construction accidents that occurred between 2017–2019; the Israeli Ministry of Environmental Protection, which provided the measures of air pollution and weather for those years; and Kav LaOved, a nonprofit organization focused on workers' rights, which provided a complementary construction site accident data.

2.1.1 Construction Sites:

The initial construction site sample provided by the Ministry of Economy and Industry included 25,571 construction sites active in Israel between 2017 and 2019. Using geo-coding techniques, I matched the sites' addresses to coordinates. Knowing each site's opening and closing days, I assigned an observation to each active day for each site, which resulted in our final sample of 24,614 sites and 10,016,000 observations.

2.1.2 Accidents:

The accident sample provided by the Ministry of Economy and Industry included 1,316 accidents during the sample period. The accidents provided by Kav La-Oved did not include site IDs matching the ministry's data. So I matched the accidents to the sites by their address instead, which resulted in an additional 31 accidents. Merging the data-set of the site's active days sample and the accidents sample, I was left with 1,164 accidents per 10,016,000 working days in construction sites.

2.1.3 Environmental Data:

Air pollution and weather data were provided by the Israeli Ministry of Environmental Protection, which reported an 8-hour average of 5-minute interval readings of NO_2 (ppb), wind strength and direction (m/sec and degrees, respectively), temperature (Celsius), humidity (%), as well as other pollutants at 173 monitoring stations throughout Israel for the sample period. The locations of monitoring stations are spread out across the country, as seen in Figure 1. Each active day in a construction site is assigned the nearest reading for each variable, where 7,199 construction sites have at least one monitoring station at a 1 km distance.

2.2 Summary Statistics:

As I've done these above actions in ArcGIS, I upload in the next chunk the finalized data-set and replicate the summary statistics.

Load libraries:

```
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(
  tidyverse,      # for data wrangling
  tidymodels,     # for ml regressions
  broom,          # convert results to tidy objects
  fastDummies,    # for turning categorical variables into sets of dummies
  fixest,         # for twfe estimation and se clustering
  hdm,            # for double lasso
  gridExtra       # for graphs

)
```

Loading raw pollution data:

```
load("Data/Process/Pollution.Rda")
```

I first create the summary statistics of the pollutants and weather variables (Table 1).

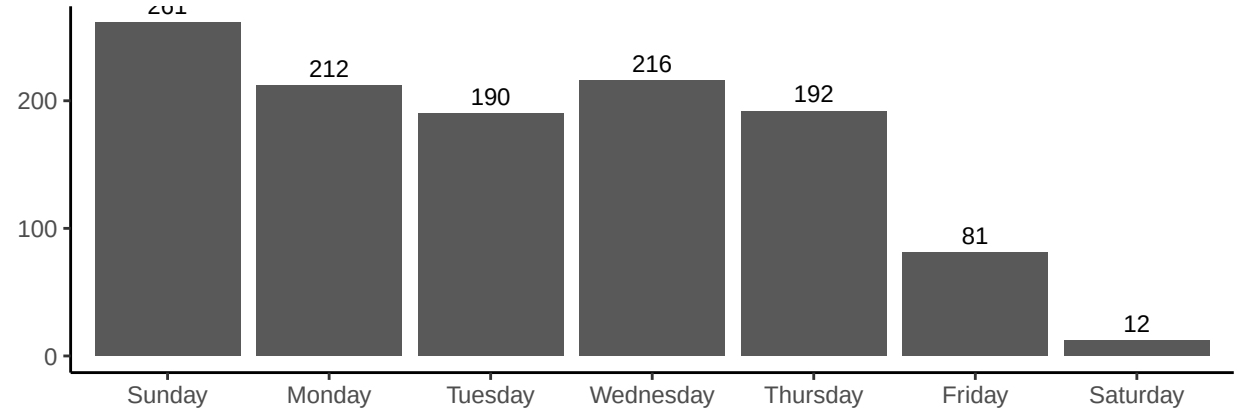
```
pollution_data_tidy <- pollution %>%
  pivot_longer(cols = c("no2_08", "no2_16", "no2_24", "pm25_08", "pm25_16", "pm25_24", "temp_08", "temp
               names_to = c("Pollutant", "Hour Measured"),
               names_sep = "_")

pollution_summary <- pollution_data_tidy %>%
  group_by(Pollutant, `Hour Measured`) %>%
```

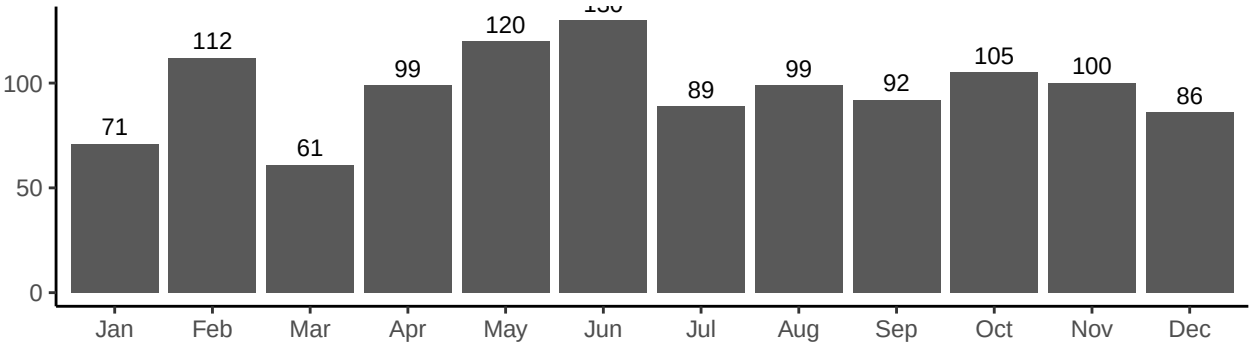
```
  summarise(Units = ifelse(grepl("no2", Pollutant), "ppb",
                    ifelse(grepl("pm", Pollutant), "µg/m3",
                    ifelse(grepl("temp", Pollutant), "Celsius",
                    ifelse(grepl("wind", Pollutant), "m/sec",
                    ifelse(grepl("humidity", Pollutant), "%",
                    ifelse(grepl("so2", Pollutant), "ppb",
                    ifelse(grepl("o3", Pollutant), "ppb",
                    ifelse(grepl("pm10", Pollutant), "µg/m3",
                    ifelse(grepl("co", Pollutant), "ppm", NA
                           ))))))))),
            Monitors = n_distinct(mon_id, na.rm = TRUE),
            Obs = sum(!is.na(value)),
            `Average Rate` = round(mean(value, na.rm = TRUE), 1),
            `Standard Error` = round(sd(value, na.rm = TRUE), 1),
            .groups = "drop_last") %>%
  arrange(Pollutant, `Hour Measured`) %>%
  unique()

knitr::kable(pollution_summary, align = "lccrr")
```

Pollutant	Hour Measured	Units	Monitors	Obs	Average Rate	Standard Error
co	08	ppm	173	16709	0.4	0.5
co	16	ppm	173	16590	0.4	0.6
co	24	ppm	173	16786	0.4	0.6
humidity	08	%	173	88684	72.4	18.5
humidity	16	%	173	91280	52.6	15.8
humidity	24	%	173	91362	66.2	17.4
no2	08	ppb	173	136492	10.9	9.9
no2	16	ppb	173	134707	10.1	18.0
no2	24	ppb	173	136697	12.4	14.9
o3	08	ppb	173	64352	27.1	13.2
o3	16	ppb	173	63993	45.9	10.7
o3	24	ppb	173	64583	36.0	11.9
pm10	08	µg/m3	173	23285	44.3	53.6
pm10	16	µg/m3	173	23021	55.6	70.6
pm10	24	µg/m3	173	23379	48.1	54.6
pm25	08	µg/m3	173	65170	20.8	12.5
pm25	16	µg/m3	173	64317	21.3	14.8
pm25	24	µg/m3	173	65343	20.6	16.0
so2	08	ppb	173	86180	0.8	0.9
so2	16	ppb	173	85921	1.2	1.7
so2	24	ppb	173	86641	0.9	1.1
temp	08	Celsius	173	111176	18.9	6.0
temp	16	Celsius	173	111156	24.7	6.4
temp	24	Celsius	173	111649	21.5	6.2
wind	08	m/sec	173	101905	1.8	1.3
wind	16	m/sec	173	101981	3.3	1.4
wind	24	m/sec	173	102253	2.3	1.2

Next I plot the distribution of the construction accidents by days of the week and months:

```r
load("Data/Final/Pollution and Accidents - Final.Rda") # Loading final data-set

data$dayofweek <- factor(data$dayofweek, levels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursda

accidents_dayweek <- aggregate(acc ~ dayofweek, data, FUN = sum) # Creating a data-frame with counts of

p1 <- ggplot(accidents_dayweek, aes(x = dayofweek, y = acc)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = acc), vjust = -0.5, size = 3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = NULL, y = NULL) +
  theme_classic() +
  scale_fill_manual(values = rainbow(7)) # Creating a bar chart for day of week:

accidents_month <- aggregate(acc ~ month, data, FUN = sum)
accidents_month$month <- month.abb[accidents_month$month] # Creating a data-frame with counts of accide


p2 <- ggplot(accidents_month, aes(x = month, y = acc)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = acc), vjust = -0.5, size = 3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = NULL, y = NULL) +
  theme_classic() +
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                              "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")) +
  scale_fill_manual(values = rainbow(3)) # Creating a bar chart for month of year


p2 <- p2 + theme(plot.margin = margin(0.5, 0, 0, 0, "in"))
p <- grid.arrange(p1, p2, ncol = 1, nrow = 2, heights = c(4, 4))
```





create variables which are important for the analysis