PSTAT 131 Final Project: Model comparison for predicting the salary of a data science job

Nicholas Axl Andrian

2023-11-23

Dataset used: https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023

In this project, we will be fitting several different machine learning algorithms to find out which method of prediction is the most accurate in getting the predicted salary(in usd).

About the data set's variables (excerpt from the kaggle site)

- work_year: The year the salary was paid.
- experience_level: The experience level in the job during the year
- employment_type: The type of employment for the role
- job_title: The role worked in during the year.
- salary: The total gross salary amount paid.
- salary_currency: The currency of the salary paid as an ISO 4217 currency code.
- salaryinusd: The salary in USD
- employee_residence: Employee's primary country of residence in during the work + year as an ISO 3166 country code.
- remote_ratio: The overall amount of work done remotely
- company_location: The country of the employer's main office or contracting branch
- company_size: The median number of people that worked for the company during the year

```
library(dplyr)
library(randomForest)
library(gbm)
library(ISLR)
library(tree)
library(tidyverse)
library(ggplot2)
library(gridExtra)


options("max.print" = 5) # to prevent page number bloat
```

Part 1: Exploratory Data Analysis

Loading the dataset

```
salaries <- read.csv("ds_salaries.csv")
```

Checking the structure of the dataset

```r
head(salaries)
```

```
##      work_year experience_level employment_type job_title salary
##      salary_currency salary_in_usd employee_residence remote_ratio
##      company_location company_size
##  [ reached 'max' / getOption("max.print") -- omitted 6 rows ]
```

```r
str(salaries)
```

```
## 'data.frame':    3755 obs. of  11 variables:
##  $ work_year         : int  2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ experience_level  : chr  "SE" "MI" "MI" "SE" ...
##  $ employment_type   : chr  "FT" "CT" "CT" "FT" ...
##  $ job_title         : chr  "Principal Data Scientist" "ML Engineer" "ML Engineer" "Data Scientist"
##  $ salary            : int  80000 30000 25500 175000 120000 222200 136000 219000 141000 147100 ...
##  $ salary_currency   : chr  "EUR" "USD" "USD" "USD" ...
##  $ salary_in_usd     : int  85847 30000 25500 175000 120000 222200 136000 219000 141000 147100 ...
##  $ employee_residence: chr  "ES" "US" "US" "CA" ...
##  $ remote_ratio      : int  100 100 100 100 100 0 0 0 0 0 ...
##  $ company_location  : chr  "ES" "US" "US" "CA" ...
##  $ company_size      : chr  "L" "S" "S" "M" ...
```

Already we can see an issue that needs to be worked on. Several variables seem to supposedly be read in as factors. We will finish conducting checks on the dataset before converting said columns.

Checking the summary of the dataset

```r
summary(salaries)
```

```
##    work_year    experience_level   employment_type     job_title
##      salary         salary_currency     salary_in_usd    employee_residence
##   remote_ratio     company_location   company_size
##  [ reached getOption("max.print") -- omitted 6 rows ]
```

Checking for null values

```r
colSums(is.na(salaries))
```

```
##        work_year experience_level  employment_type        job_title
##                0                0                0                0
##           salary
##                0
##  [ reached getOption("max.print") -- omitted 6 entries ]
```

Fortunately, we have no null values so imputing is not required

Checking potential factor columns for their unique values

```r
factor_cols <- salaries[, c(2, 3, 4, 6, 8, 10, 11)]

# finding unique values, referenced code from https://www.kaggle.com/code/abdulfaheem11/data-science-sa

# output ommitted to prevent too much space being taken up
sapply(factor_cols, function(col) unique(col))
```

```
## $experience_level
## [1] "SE" "MI" "EN" "EX"
##
## $employment_type
## [1] "FT" "CT" "FL" "PT"
##
## $job_title
## [1] "Principal Data Scientist" "ML Engineer"
## [3] "Data Scientist"           "Applied Scientist"
## [5] "Data Analyst"
##  [ reached getOption("max.print") -- omitted 88 entries ]
##
## $salary_currency
## [1] "EUR" "USD" "INR" "HKD" "CHF"
##  [ reached getOption("max.print") -- omitted 15 entries ]
##
## $employee_residence
## [1] "ES" "US" "CA" "DE" "GB"
##  [ reached getOption("max.print") -- omitted 73 entries ]
##
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

Changing said variables to become factors

```
salaries[, c(2, 3, 4, 6, 8, 10, 11)] <- lapply(factor_cols, factor)
str(salaries)
```

```
## 'data.frame':    3755 obs. of  11 variables:
##  $ work_year         : int  2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ experience_level  : Factor w/ 4 levels "EN","EX","MI",..: 4 3 3 4 4 4 4 4 4 4 ...
##  $ employment_type   : Factor w/ 4 levels "CT","FL","FT",..: 3 1 1 3 3 3 3 3 3 3 ...
##  $ job_title         : Factor w/ 93 levels "3D Computer Vision Researcher",..: 85 78 78 48 48 9 9 48
##  $ salary            : int  80000 30000 25500 175000 120000 222200 136000 219000 141000 147100 ...
##  $ salary_currency   : Factor w/ 20 levels "AUD","BRL","CAD",..: 8 20 20 20 20 20 20 20 20 20 ...
##  $ salary_in_usd     : int  85847 30000 25500 175000 120000 222200 136000 219000 141000 147100 ...
##  $ employee_residence: Factor w/ 78 levels "AE","AM","AR",..: 27 76 76 12 12 76 76 12 12 76 ...
##  $ remote_ratio      : int  100 100 100 100 100 0 0 0 0 0 ...
##  $ company_location  : Factor w/ 72 levels "AE","AL","AM",..: 26 71 71 13 13 71 71 13 13 71 ...
##  $ company_size      : Factor w/ 3 levels "L","M","S": 1 3 3 2 2 1 1 2 2 2 ...
```

We can also drop the salary as we will just be using the salary_in_usd to simplify our steps.

```
salaries <- salaries[, !(names(salaries) %in% c('salary_currency','salary'))]
```

Visualization to search for patterns with regards to the salary_in_usd

Prioritizing focus on work_year, experience_level, employment_type, job_title, employee_residence, remote_ratio, company_location, company_size

```
yearplot <- ggplot(salaries, aes(x = work_year, y = salary_in_usd)) +
  geom_point(color = "red", size = 3) +
  labs(x = "Work Year", y = "Salary in USD", title = "Salary vs Work Year")
```

```r
# Boxplot using ggplot
expplot <- ggplot(salaries, aes(x = experience_level, y = salary_in_usd)) +
  geom_boxplot(fill = "skyblue") +
  labs(x = "Experience Level", y = "Salary in USD", title = "Salary vs Experience Level")

grid.arrange(yearplot, expplot, ncol = 2)
```

- We can see that the average salary in usd increases as the years go by, as the line congests further upwards towards the end.
- Experience level does not really show much of a trend as it goes towards seniority, We can tell though that EX has the highest average and MI has the highest peak

```r
employplot <- ggplot(salaries, aes(x = employment_type, y = salary_in_usd)) +
  geom_boxplot(fill = "skyblue") +
  labs(x = "Employment Type", y = "Salary in USD", title = "Salary vs Employment Type")

remoteplot <- ggplot(salaries, aes(x = remote_ratio, y = salary_in_usd)) +
  geom_point(color = "red", size = 3, shape = 19) +
  labs(x = "Remote Ratio", y = "Salary in USD", title = "Salary vs Remote Ratio")

grid.arrange(employplot, remoteplot, ncol = 2)
```

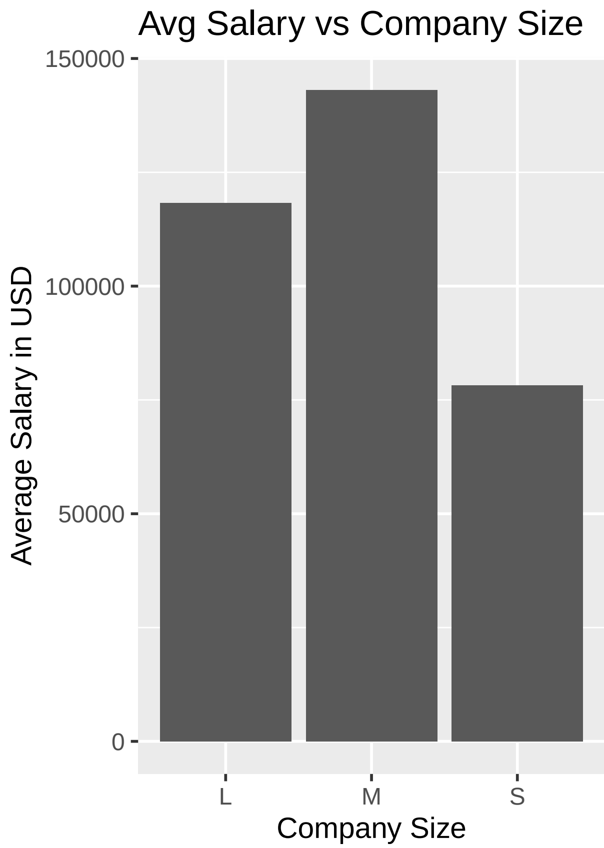
- In employment type, FT has the highest average as well as higher peaks
- Remote ratio consists of 0, 50 and 100. The highest points as well as average are in the order 0>100>50

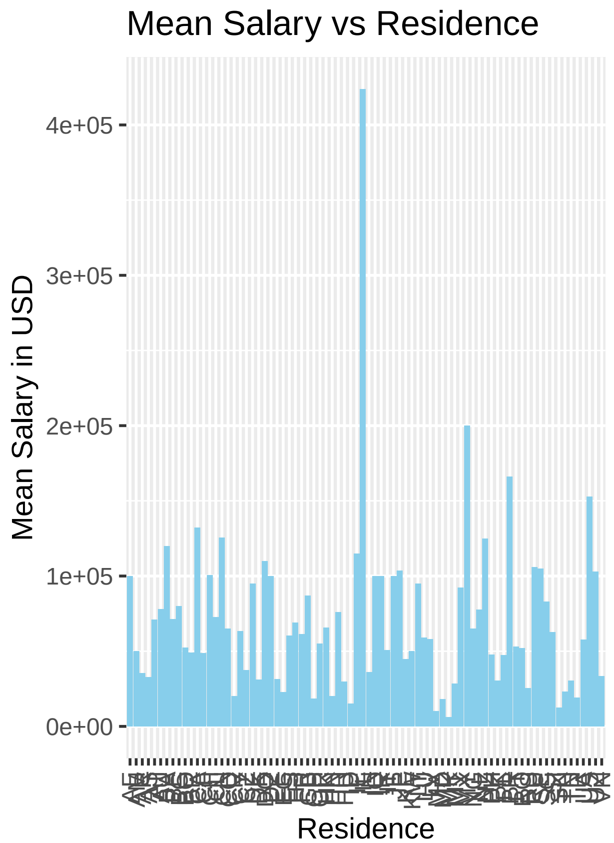
```
usd_salary_by_size <- salaries%>%
  group_by(company_size)%>%
  summarise(Avg_sal=mean(salary_in_usd))

sizeplot <- ggplot(usd_salary_by_size, aes(x=company_size, y=Avg_sal)) +
  geom_col() +
  labs(title='Avg Salary vs Company Size', x='Company Size', y='Average Salary in USD')

residenceplot <- ggplot(salaries, aes(x = employee_residence, y = salary_in_usd)) +
  geom_bar(stat = "summary", fun = "mean", fill = "skyblue") +
  labs(x = "Residence", y = "Mean Salary in USD", title = "Mean Salary vs Residence") + theme(axis.text

grid.arrange(sizeplot,residenceplot, ncol = 2)
```





+ From this plot we can also see that medium sized companies pay the largest on average, followed by large then small + We can see that out of all the residences, IL has the largest mean salary by a huge margin. This may be alarming so we will have to keep an eye on it as it may be an inaccurate input

Plotting job_title/employee_residence/company_location would be way too congested due to overwhelming amounts of factor levels. I have decided to just show a summary of their statistics

```
title_summary <- salaries %>%
  group_by(job_title) %>%
  summarise(
    mean_salary = mean(salary_in_usd),
    median_salary = median(salary_in_usd),
    min_salary = min(salary_in_usd),
    max_salary = max(salary_in_usd),
    Q1 = quantile(salary_in_usd, probs = 0.25),
    Q3 = quantile(salary_in_usd, probs = 0.75)
  )
residence_summary <- salaries %>%
  group_by(employee_residence) %>%
  summarise(
    mean_salary = mean(salary_in_usd),
    median_salary = median(salary_in_usd),
    min_salary = min(salary_in_usd),
    max_salary = max(salary_in_usd),
    Q1 = quantile(salary_in_usd, probs = 0.25, na.rm = TRUE),
    Q3 = quantile(salary_in_usd, probs = 0.75, na.rm = TRUE)
  )
```

```r
location_summary <- salaries %>%
  group_by(company_location) %>%
  summarise(
    mean_salary = mean(salary_in_usd),
    median_salary = median(salary_in_usd),
    min_salary = min(salary_in_usd),
    max_salary = max(salary_in_usd),
    Q1 = quantile(salary_in_usd, probs = 0.25, na.rm = TRUE),
    Q3 = quantile(salary_in_usd, probs = 0.75, na.rm = TRUE)
  )
head(title_summary)
```

```
## # A tibble: 6 x 7
##   job_title        mean_salary median_salary min_salary max_salary     Q1     Q3
##   <fct>                  <dbl>         <dbl>      <int>      <int>  <dbl>  <dbl>
## 1 3D Computer Vis~      21352.         15000       5409      50000 8.85e3 2.75e4
## 2 AI Developer         136666.        108000       6304     300000 6.97e4 2.07e5
## 3 AI Programmer         55000          55000      40000      70000 4.75e4 6.25e4
## 4 AI Scientist         110121.         52500      12000     423834 3.11e4 2   e5
## 5 Analytics Engin~     152369.        143860       7500     289800 1.17e5 1.85e5
## 6 Applied Data Sc~     113726.         74159      20670     380000 5.11e4 1.45e5
```

```r
head(residence_summary)
```

```
## # A tibble: 6 x 7
##   employee_residence mean_salary median_salary min_salary max_salary     Q1
##   <fct>                    <dbl>         <dbl>      <int>      <int>  <dbl>
## 1 AE                      100000        115000      65000     120000 90000
## 2 AM                       50000         50000      50000      50000 50000
## 3 AR                       35500         39000      12000      60000 17250
## 4 AS                       32778.        32778.     20000      45555 26389.
## 5 AT                       71126.        68060.     50000      91237 60567
## 6 AU                       77981.        75050      40000     150000 49209
## # i 1 more variable: Q3 <dbl>
```

```r
head(location_summary)
```

```
## # A tibble: 6 x 7
##   company_location mean_salary median_salary min_salary max_salary     Q1     Q3
##   <fct>                  <dbl>         <dbl>      <int>      <int>  <dbl>  <dbl>
## 1 AE                    100000        115000      65000     120000 90000  117500
## 2 AL                     10000         10000      10000      10000 10000   10000
## 3 AM                     50000         50000      50000      50000 50000   50000
## 4 AR                     25000         13000      12000      50000 12500   31500
## 5 AS                     29351         20000      18053      50000 19026.  35000
## 6 AT                     71355.        68060.     50000      91237 61598.  85512
```

Part 2: Problem formulation and preparation for statistical learning algorithms