Factors Impacting Number of LinkedIn Job Applications per Post View

JRLK - Jessie Ringness, Rebekah Kim, Laura Cai, Karen Dong

2023-11-14

Introduction and data

LinkedIn is a popular platform that connects companies and professionals spanning various levels of experience, and there are thousands of active job postings available on LinkedIn. Moreover, online job search services and platforms are now considered equally important for people to access a wide variety of opportunities compared to in-person job postings. With the sheer amount of postings, applicants may be overwhelmed by the vast amount of postings, and are less likely to come across some postings over others. Our primary research question is - what variables about job postings increase popularity among applicants?

This data set was created by Arsh Koneru-Ansari in July 2023, who used Python to scrape data directly from linkedin.com. The scraper code is published in their GitHub.

The data dictionary for the variable definitions can be found in the ReadMe for the data. The variables we will focus on are:

- **`applies`:** number of applications that have been submitted
- **`views`:** number of times the job posting has been viewed
- **`max_salary`:** maximum salary offered in the position
- **`remote_allowed`:** whether job permits remote work (1 = yes)
- **`follower_count`:** number of company followers on LinkedIn
- **`listed_time`:** time when the job was listed, in UNIX time
- **`formatted_experience_level`:** job experience level (entry level, associate, mid-senior level, director, executive, internship)
- **`type`**: type of benefit provided (Medical insurance, Dental insurance, 401(k), Paid maternity leave, Disability insurance, Vision insurance, Tuition assistance, Pension plan, Child care support, Commuter benefits, Student loan assistance)

Another potential interaction effect is the number of views and experience level. Different positions are more sought after on LinkedIn over others, depending on viewers' backgrounds. The number of applications increases per number of views for a position requiring associate experience increases at a more rapid rate than those for other positions, although all positions have some sort of impact associated with views and applications.

Fig 1.1. The distribution of hourly maximum salary is mostly concentrated when the maximum hourly salary is between 0 and 150. It is right skewed and there are apparent outliers, when the maximum salary is above 200.

Fig 1.2. The distribution of the hour listed is concentrated mainly between hours 15 and 20. This distribution is left skewed and there are outliers between hours 0 and 5.

Fig 1.3. The distribution of follower count is also right skewed, with a peak around 0. There are apparent outliers when follower count is greater than 5.0e+06.

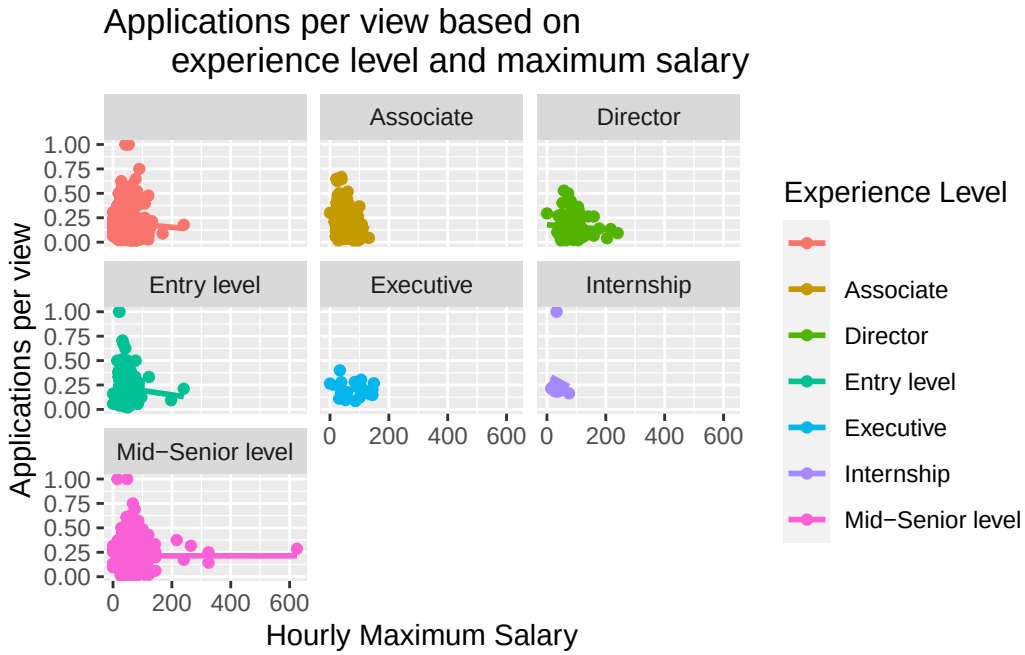

Fig 2. The distribution of applications per view based on experience level and maximum salary is mostly concentrated when the applications per view is less than 0.50 and the hourly maximum salary is less than 200 for each experience level. The mid-senior level has apparent outliers when the hourly maximum salary is greater than 200 and the NA level has outliers when the applications per view is above 0.75. There is also not as much data for some of the experience levels, including internship and executive.

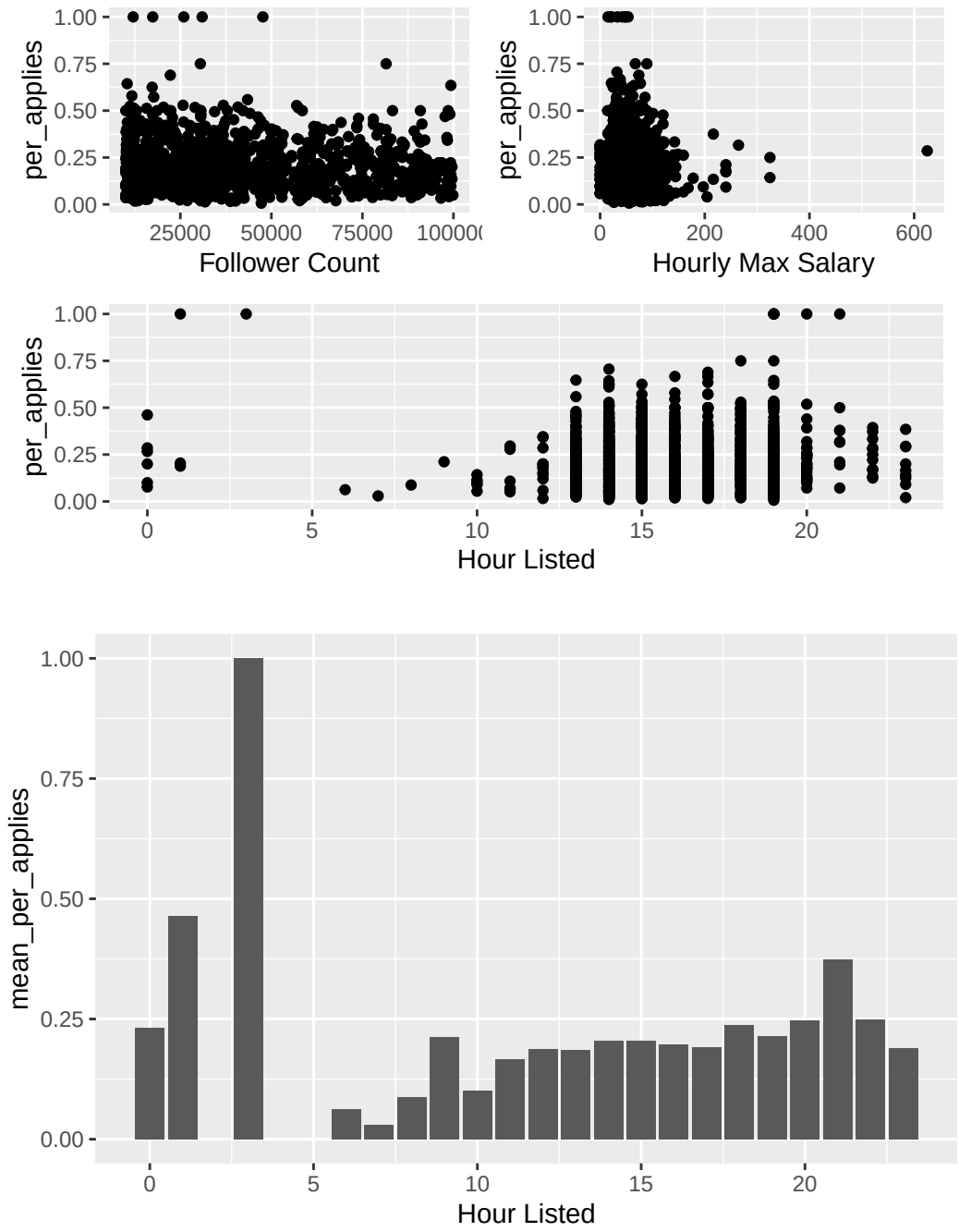

Fig 3.1. There is a slight linear correlation between follower count of a company and percentage of viewers who apply to the job. Most of the observations are concentrated to have less than 1,000,000 followers, so it would be beneficial to remove outliers that have more followers than that.

Fig 3.2. There is a slight positive linear correlation between a job's adjusted hourly maximum salary and percentage of viewers who apply to the job. Most of the observations are concentrated to have less than $200,000 for adjusted hourly max salary, so it would be beneficial to filter out outliers that are above this threshold. Additionally, it would be beneficial to remove the outlier where 100% of viewers applied to the job (per_applies = 1.0)

Fig 3.3. Convert to categorical variable, make a histogram that's colored by morning, afternoon, evening

Fig 3.4. This figure could be statistically misleading since the intervals are not consistent, and mean_per_applies is not a helpful response variable for the model. We should convert hour_listed to a categorical variable and create a pie chart instead to show the frequency of each level.

Methodology

While the bulk of our data was found in the 'job_postings' data set, we also wanted to include employee count and follower count data in the 'employee' data set and the type and number of benefits listed on the post found in the 'benefits' data set. We needed to manipulate the data in 'benefits' to create a useful predictor as the majority of the data was NA, meaning no benefits could be found from the scraped data. To make the data from 'benefits' a useful predictor, we created a new categorical variable called ''if_benefits' where if a benefit (such as paid maternity leave or a 401k plan) was listed on the post, then the post was considered as having benefits 'listed', and otherwise was listed as 'none' listed. We joined all data sets together by 'company_id', and saved the data set as 'linkedin'.

We also made some assumptions about other variables to normalize predictor variables. The categorical variable 'pay_period' contained data on when the job would pay its worker the 'max_salary' or 'min_salary' amount, with hourly, monthly, and annual payments as the different levels. To normalize the 'max_salary' amount, we calculated the hourly wage given the maximum pay for hourly, monthly, and yearly pay periods. We assumed 160 hours for the monthly payments (40 hour work week for 4 weeks), and 2080 hours for the annual payments (40 hour work week for 52 weeks). We saved the new data in a variable called 'hourly_max_salary'.

We then dropped all 'NA' values for all predictors we wanted to observe so that we could keep our dataset consistent when testing different models, meaning NA values for 'hourly_max_salary', 'per_applies', 'follower_count', 'formatted_experience_level', 'original_listed_time', and 'remote_allowed'.

Because the number of views an application gets is directly related to the number of applications and the jobs have been listed for varying durations of time, we decided to normalize the number of applications with the views. To do so, we created a new variable 'per_applies'. We then used 'per_applies' as our response variable.

Because 'per_applies' is a numerical variable, a linear regression model would be most appropriate to predict the number of applications per view. As we addressed in the introduction, a person takes into consider many different factors when applying to a job, so our model takes into consideration multiple predictors, including the hour the job was posted, the number of followers the company has, the job experience level, the maximum salary, ability to work remote, and if benefits are listed.

We split the 'linkedin' dataset into training and testing data, with 75% of the data in training and 25% in testing. We then used cross-fold validation with 12 folds on the training data set to find the mean summary statistics (AIC, BIC, Adjusted R-Squared) for each model and compared the different values to find the best possible model. This process was repeated for models containing each combination of predictor variables. We set a seed of (2) when splitting and folding the data to ensure reproducibility.

```
set.seed(2)
calc_model_stats <- function(x) {
  glance(extract_fit_parsnip(x)) |>
    select(adj.r.squared, AIC, BIC)
}
```

```
map_df(linkedin_fit_rs_full$.extracts, ~ .x[[1]][[1]]) |>
  summarise(mean_adj_rsq = mean(adj.r.squared),
            mean_aic = mean(AIC),
            mean_bic = mean(BIC))
```

```
# A tibble: 1 x 3
  mean_adj_rsq mean_aic mean_bic
         <dbl>    <dbl>    <dbl>
1       0.0251   -1518.   -1451.
```

```
map_df(linkedin_fit_rs_red$.extracts, ~ .x[[1]][[1]]) |>
  summarise(mean_adj_rsq = mean(adj.r.squared),
            mean_aic = mean(AIC),
            mean_bic = mean(BIC))
```

```
# A tibble: 1 x 3
  mean_adj_rsq mean_aic mean_bic
         <dbl>    <dbl>    <dbl>
1      0.00989   -1507.   -1486.
```

The adjusted R^2 value for a model including `hourly_max_salary`, `follower_count`, `remote_allowed`, `formatted_experience_level`, `hour_listed`, and `if_benefits` was 0.0076. We removed variables one by one, but each model resulted in a lower R^2 value, indicating that a model including all mentioned variables is the best.

```
linkedin_fit_full <- linkedin_wflow1 |>
  fit(data = linkedin_test)

tidy(linkedin_fit_full) |>
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.132	0.041	3.246	0.001
hourly_max_salary	0.000	0.000	0.461	0.645
follower_count	0.000	0.000	-0.264	0.792
hour_listed	0.005	0.002	1.947	0.052
remote_allowed_X1	0.043	0.015	2.808	0.005
formatted_experience_level_Associate	-0.005	0.019	-0.263	0.793
formatted_experience_level_Director	-0.072	0.028	-2.569	0.011
formatted_experience_level_Entry.level	0.009	0.019	0.462	0.644
formatted_experience_level_Executive	-0.036	0.076	-0.477	0.634
formatted_experience_level_Internship	-0.043	0.075	-0.576	0.565
formatted_experience_level_Mid.Senior.level	-0.009	0.016	-0.567	0.571
if_benefits_listed	-0.020	0.012	-1.622	0.105

Using a significance level of $\alpha = 0.10$, `hourly_max_salary`, `follower_count`, and if the job requires internship or director level of experience are the only statistically significant variables, with p-values of 0.093, 0.0183, 0.012, and 0.003 respectively. A smaller model using `hourly_max_salary`, `follower_count`, and `job_experience` was made.

```
#linkedin_test_pred_full <- predict(linkedin_fit_full, linkedin_train) |>
  #bind_cols(linkedin_train)
#map_df(linkedin_fit_full$.extracts, ~ .x[[1]][[1]]) |>
  #summarise(mean_adj_rsq = mean(adj.r.squared),
            #mean_aic = mean(AIC),
            #mean_bic = mean(BIC))

#linkedin_test_pred_red <- predict(linkedin_fit_red, linkedin_train) |>
  #bind_cols(linkedin_train)
#rsq(linkedin_test_pred_red, truth = per_applies, estimate = .pred)
```

Results

```
linkedin_yr_rec <- recipe(per_applies ~ job_id + hourly_max_salary + follower_count + remo
  update_role(job_id, new_role = "ID") |>
  step_naomit(all_predictors()) |>
  step_dummy(all_nominal_predictors()) |>
  step_zv(all_predictors())
```

```
#specify the model
linkedin_yr_spec <- linear_reg() |>
  set_engine("lm")

#build model workflow
linkedin_yr_workflow <- workflow() |>
  add_model(linkedin_yr_spec) |>
  add_recipe(linkedin_yr_rec)

# fit the model
linkedin_yr_fit <- linkedin_yr_workflow |>
  fit(data = linkedin_subset)

tidy(linkedin_yr_fit) |>
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.181	0.021	8.664	0.000
hourly_max_salary	0.000	0.000	-2.392	0.017
follower_count	0.000	0.000	-0.892	0.372
hour_listed	0.002	0.001	1.207	0.227
remote_allowed_X1	0.039	0.008	5.053	0.000
formatted_experience_level_Associate	0.014	0.010	1.380	0.168
formatted_experience_level_Director	-0.031	0.015	-2.030	0.042
formatted_experience_level_Entry.level	0.021	0.010	1.994	0.046
formatted_experience_level_Executive	0.011	0.032	0.349	0.727
formatted_experience_level_Internship	0.095	0.048	1.992	0.047
formatted_experience_level_Mid.Senior.level	0.018	0.008	2.169	0.030
if_benefits_listed	-0.004	0.006	-0.574	0.566

```
glance(linkedin_yr_fit)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic     p.value    df logLik    AIC    BIC
      <dbl>         <dbl> <dbl>     <dbl>       <dbl> <dbl>  <dbl>  <dbl>  <dbl>
1    0.0280        0.0225 0.134      5.01     9.61e-8    11  1142. -2258. -2186.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Looking at the how the maximum salary, follower count, whether the job allows remote, the experience level, if there were benefits listed, and the hour at which the job was listed all affected the ratio of applications to views, there is no significant relationship between these variables, since the adjusted R^2 s 0.0589, which is very low. This means that about 6% of the variation in the percent of viewers that applied is a result of the variation in the predictor variables we are studying, indicating no relationship.We tried different combinations of variables to see if there was any relationship between the variables with percent of viewers that applied.

Discussion + Conclusion

While we had expected the percent of viewers that applied for each job to increase as the maximum salary and number of followers increase, our model and EDA have demonstrated how there is no relationship between these variables, along with having a remote option, the job experience level, the hour at which it was posted, and if the benefits were posted, and our response variable of the percent of viewers who applied.

Some of the limitations with our analysis may include assumption errors with the number of hours the job would work for the jobs listed as yearly and monthly, the filtering of outliers and deciding the thresholds for outliers, model complexity where we are fitting many predictor variables, and the lack of data considering how long the jobs were listed.

Some ways our analysis could be improved would be filtering out the outliers better, deciding the thresholds for views, follower count, salaries, etc, that would avoid skewing the data.

Potential issues relating to the data would include the time scraped from LinkedIn, since many of the original listed times and the listed times (the time it was scraped) is exactly the same, which is not meaningful in determining the number of applications over a certain amount of time.