7/31/2022

Tran Minh Khoa

2101432I

MLDP Project

Report

# Introduction

## Topic: Data Science Job Salaries

Jobs in the field of Data Science vary widely in salary. This project aims to determine how various factors – such as job title, company size, and country of work – impact annual salary.

## Dataset

**URL:** <https://www.kaggle.com/datasets/ruchi798/data-science-job-salaries>

**Columns:**

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| company\_location | Nominal | Country Code1 of company’s contracting branch (50 unique values) |
| company\_size | Ordinal | Number of people that worked for the company  *With being the number of employees,*   |  |  | | --- | --- | | Value | Meaning | | S |  | | M |  | | L |  | |
| employee\_residence | Nominal | Country Code1 of employee’s primary residence (57 unique values) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| experience\_level | Ordinal | Level of experience holding *job\_title*   |  |  | | --- | --- | | Value | Meaning | | EN | Entry-level / Junior | | MI | Mid-level / Intermediate | | SE | Senior-level / Expert | | EX | Executive-level / Director | |
| job\_title | Nominal | Role of work (50 unique values) |
| remote\_ratio | Ordinal | Percentage of work done remotely  *With being the percentage of remote work,*   |  |  | | --- | --- | | Value | Meaning | | 0 |  | | 50 |  | | 100 |  | |
| salary\_currency | Nominal | Currency of *salary* |
| salary | Interval | The salary paid |
| salary\_in\_usd | Interval | *salary* converted to USD |
| work\_year | Interval | Year when salary was paid |

1 [ISO 3166](https://www.iso.org/iso-3166-country-codes.html)

2 [ISO 4217](https://www.iso.org/iso-4217-currency-codes.html)

# Data Exploration & Pre-processing

## Preprocessing

### Cleaning

|  |  |  |
| --- | --- | --- |
| Issue | Description | Fix(es) |
| Duplicates | There were no duplicate indexes, but 71 rows with duplicate values.  These duplicates were sparsely distributed (usually 2 per unique duplicate). | *Ignore*  They probably refer to employees working for the same company, in the same role. |
| salary & `salary\_currency` | These are used to calculate, and hence directly related to, `salary\_in\_usd`. | *Drop*  `salary\_in\_usd` can accurately represent them. |
| Low Variance columns | `employment\_type` is dominated by “FT” (96.9% of values) | *Drop* |
| Missing values | There were no missing values | |

### Transformation

The general motive is to reduce spread & numericize for better [Modeling](#_Methods_&_Improvements).

|  |  |
| --- | --- |
| Transformation | Justification |
| *Integer-ize* `company\_size` | Ordinal columns can be Numericized as rank numbers |
| *Integer-ize* `experience\_level` |

|  |  |
| --- | --- |
| *Boolean-ize* `employee\_residence`  *Rename* as *`ricl`* (*r*esides *i*n *c*ompany *l*ocation) | Has many similarities with `company\_location`; it would be more helpful to know, not where the employee lives, but whether they live in the same country as their company. |
| *Bin* `company\_location` by continent | Numerosity Reduction of 50 unique values to 6 |
| *Integer-ize* `company\_location` from West to East   * The Westmost, NA, is 1 * The Eastmost, OC, is 6 | Attempt at Numericizing |
| *Standardize* `job\_title` with the following [RegEx](https://en.wikipedia.org/wiki/Regular_expression):   1. /.\*Researcher/ ⇒ “Research Scientist” 2. /AI|Computer Vision|Machine Learning|NLP/ ⇒ “ML” 3. /.\*(Data|ML).\*(Analyst|Architect|Engineer|Researcher|Scientist).\*/ ⇒ “$1 $2” 4. /(Director|Head).\*|.\*Manager/ ⇒ “Manager”   *Bin* titles with less than 5% frequency as “Other” | Numerosity Reduction of 50 unique values to 6;  Many titles had similarities, only different due to prefixes/suffixes or acronyms, etc.  This approach assumes that the standardized titles are truly similar. |

|  |  |
| --- | --- |
| *Popularity Encode* `job\_title`  *Rename* as `job\_popularity` | (Singh, 2020)  Attempt at Numericizing;  Better than *Integer Encoding*, which gives false value to the difference between titles |
| *Min-Max Normalize* all numeric columns except `work\_year` | [Modeling](#_Methods_&_Improvements) will assign weights. Normalizing allows weights to be interpreted on the same scale (as significances)  Years outside this dataset should still be able to be passed into models, so Min-Max scaling would not make sense on `work\_year`. |

### Further Cleaning

|  |  |  |
| --- | --- | --- |
| Issue | Description | Fix(es) |
| Outliers | Values that are over 2 standard deviations away from the mean | *Drop*  Outliers sway modeling |
| Columns with Strong Correlation | There are no strongly correlated columns | |

## Exploration

# III & IV. Methods & Improvements + Results & Analysis

## Setup

The dataset was split into the following:

* Features: work\_year, experience\_level, remote\_ratio, company\_location, company\_size, job\_popularity, ricl
* Target: salary\_in\_usd
  + Is right-skewed, but logging makes it left-skewed.
  + Models can be trained to target both original and logged values; the better result will be selected.
  + Predictions from models using logged values will be re-exponentiated
* Test Size: 1/3

The Cost Function (Evaluation Criteria) used for all models is Squared Error (SE), presented as Root Mean SE (RMSE).

Since the Target is a continuous variable, Classifiers were not used.

## Exploration of Models + Backward Feature Elimination (BFE)

Sequential Feature Selection (BFE) was used to remove features to minimize SE.

* Results using between all and half of features are compared. Using less than half of features was tested to be ineffective and assumes that most features are irrelevant to the Target.
* The selected features which yielded the lowest (best) RMSE are shown.

### Starting with Ordinary Least Squares

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Did the Original or Logged Target give better results? | Best RMSE + Removed Features (if any) | Description |
| Linear Regression (LR) | Logged | 40608.50 | Fits a straight line which yields the Lowest Squared Error (Least Squares) |
|  | | | |

### Does Regularization improve performance?

|  |  |  |  |
| --- | --- | --- | --- |
| Ridge | Logged | 40788.88 | Introduces a bias of to LR’s Least Squares. (StatQuest with Josh Starmer, 2018a)  Useful for desensitizing the best fit line to training data. In this case, Ridge performs “worse” than LR because the train & test data are similar. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Lasso | Original | 42093.98  remote\_ratio, work\_year, & job\_popularity | Introduces a bias of to LR’s Least Squares. (StatQuest with Josh Starmer, 2018b)  Lasso performs worse than LR for the same reason as Ridge.  Ridge can only scale features according to their importance, but Lasso can also remove “useless” features. In this case, Lasso performs worse than Ridge because feature selection has already been done beforehand. |
|  | | | |

### Can Non-linear models perform better?

|  |  |  |  |
| --- | --- | --- | --- |
| Regression Tree (RT) | Logged | 43369.06  job\_popularity, work\_year, company\_size, & remote\_ratio | Partitions data based on certain rules & predicts using partition means (StatQuest with Josh Starmer, 2019b) |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Gradient Boosting Regression (GBR) | Original | 40069.15 | Chains multiple RTs to reduce error: the error of the current model is used as the target of the next. (StatQuest with Josh Starmer, 2019a) |
|  | | | |
| Support Vector Regression (SVR) | Logged | 40292.60  work\_year | Fits a hyperplane (Raj, 2020) |
|  | | | |

## Hyperparameter Tuning

The top 3 models – GBR, SVR, & LR – were shortlisted for further tuning. The optimizing algorithm used is Grid Search.

|  |  |  |
| --- | --- | --- |
| Model | Tested Parameters | Best RMSE + Parameters |
| GBR | |  |  |  | | --- | --- | --- | | Parameter | Values | Justification | | learning\_rate (Limits how much each tree contributes to the overall prediction) | 0.1, 0.01 | Prevents overfitting | | n\_estimators  (Limits number of trees) | 30-70; interval of 10 | | max\_leaf\_nodes  (Limits number of unique predictions per tree) | 4-8; interval of 1 | | tol  (Stops boosting when SE does not improve by at least this value) | 0.1-0.3; interval of 0.05 | | 40415.86   |  | | --- | | 0.1 | | 4 | | 60 | | 0.1 | |
|  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SVR | |  |  |  | | --- | --- | --- | | Parameter | Values | Justification | | tol  (Stops boosting when SE does not improve by at least this value) | 0.12-0.15; interval of 0.005 | Prevents overfitting | | C  (Introduces a bias to SE, reducing sensitivity towards training data) | 0.6-0.9; interval of 0.05 | | epsilon  (Prevents a penalty from being applied to values with SE within this value) | 0.1-0.5;  interval of 0.1 | Prevents underfitting | | 39685.80   |  | | --- | | 0.145 | | 0.75 | | 0.2 | |
|  | | |
| LR | No parameters to optimize | |

## The Best Model

SVR

Parameters

* C: 0.145
* epsilon: 0.2
* tol: 0.145

Trained with no `work\_year` and a logged Target

RMSE: 39685.80

# Conclusion

# References

Raj, A. (2020, October 3). Unlocking the True Power of Support Vector Regression. Medium. <https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0>

Singh, S. (2020, February 24). *Categorical Variable Encoding Techniques - Analytics Vidhya*. Medium. <https://medium.com/analytics-vidhya/categorical-variable-encoding-techniques-17e607fe42f9>

StatQuest with Josh Starmer. (2018a, September 24). *Regularization Part 1: Ridge (L2) Regression* [Video]. YouTube. <https://www.youtube.com/watch?v=Q81RR3yKn30>

StatQuest with Josh Starmer. (2018b, October 1). *Regularization Part 2: Lasso (L1) Regression* [Video]. YouTube. <https://www.youtube.com/watch?v=NGf0voTMlcs>

StatQuest with Josh Starmer. (2019a, March 25). *Gradient Boost Part 1 (of 4): Regression Main Ideas* [Video]. YouTube. <https://www.youtube.com/watch?v=3CC4N4z3GJc>

StatQuest with Josh Starmer. (2019b, August 20). *Regression Trees, Clearly Explained!!!* [Video]. YouTube. <https://www.youtube.com/watch?v=g9c66TUylZ4>