10/20/2021

Regression Analysis – Total reward

Olivier FONTAINE

TABLE OF CONTENTS

ANALYSIS OBJECTIVE

➔ Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation.

The objective of this analysis is to find the features that are the most important to explain total reward in a company. So, here we are focusing on interpretation rather than prediction.

DATA SET DESCRIPTION

➔ Brief description of the data set and a summary of its attributes

The data set name is *Glassdoor Gender Pay Gap*. "The data set has been taken from Glassdoor and focuses on income for various job titles, gender…". The data set is available on KAGGLE: https://www.kaggle.com/nilimajauhari/glassdoor-analyze-gender-pay-gap?select=Glassdoor+Gender+Pay+Gap.csv

Note: I selected this dataset on Gender Pay Gap but I use it for total rewards interpretation rather than gender pay interpretation.

The data set has **1000 rows** and **9 columns**.

The column **BasePay** is the target variable. But, actually the column **Bonus** can also be a target variable. It might be a good idea to create a new feature **TotalPay = BasePay + Bonus**. This will be investigated in exploratory data analysis and feature engineering part.

Here is the list of columns:

- The **Education** column is categorical at first sight but it does correspond to an **ordinal** data as High school < College < Master < PhD.
  This will be managed in the encoding section.
- The **Education** column is ordinal with lower performance at 1 and greater performance at 5.

Column	Features	Data type	Data type	Values
0	JobTitle	object	Categorical	10 different values
1	Gender	object	Categorical	'Male' 'Female'
2	Age	int64	Numerical	In years
3	PerfEval	int64	Ordinal	1 2 3 4 5
4	Education	object	Categorical	'High school' 'College' 'Master' 'PhD'

Column	Features	Data type	Data type	Values
5	Dept	object	Categorical	5 different values
6	Seniority	int64	Numerical	In years
7	**BasePay**	int64	Numerical	Annual salary, probably in USD
8	**Bonus**	int64	Numerical	Annual bonus, probably in USD

DATA EXPLORATION, DATA CLEANING & FEATURE ENGINEERING

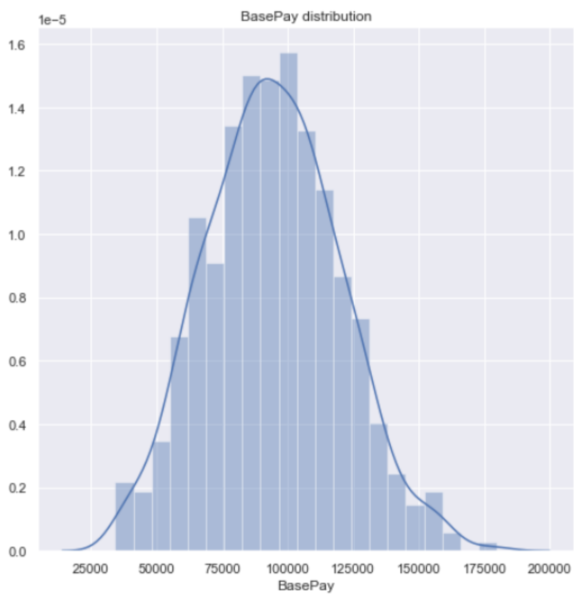
➔ Brief summary of data exploration and actions taken for data cleaning and feature engineering.

TARGET VARIABLE

BasePay is the expected target variable and is numerical.

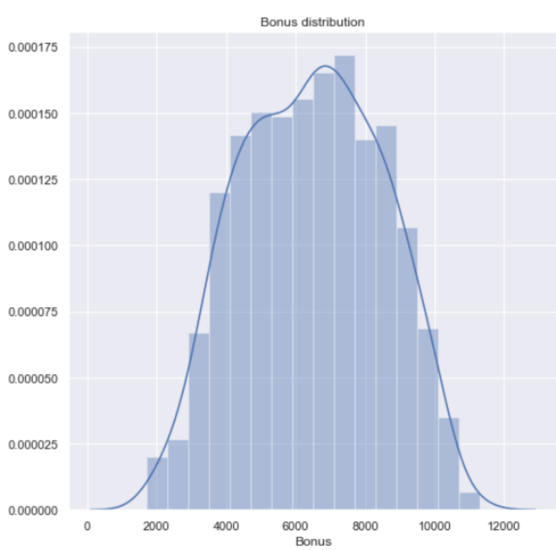
Thus, the problem is a **regression** problem.

Let's see now the distribution of **BasePay** in the company:

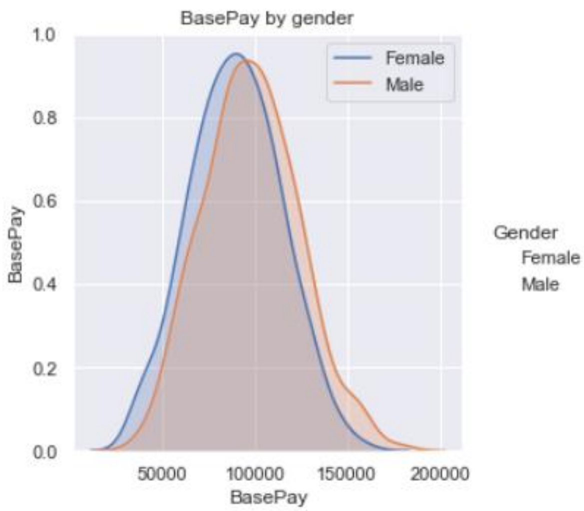


This distribution looks like a normal distribution.

See also the Bonus distribution:

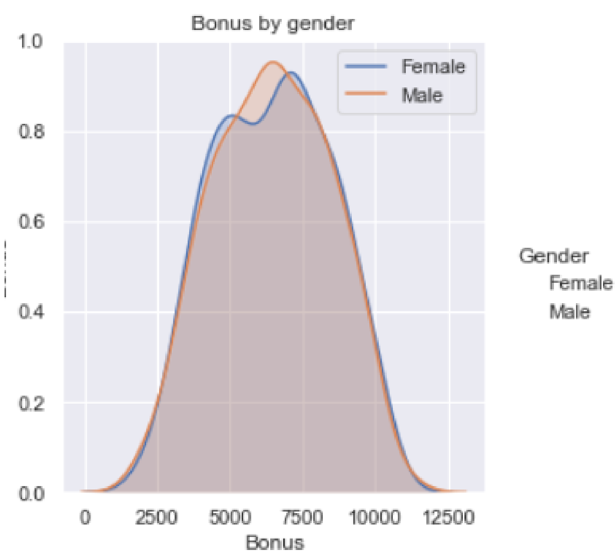


Now, let's have a look to the **BasePay** distribution by gender:



For both **BasePay**, in average, male are more paid than female.

Now, let's look at the Bonus distribution by gender:

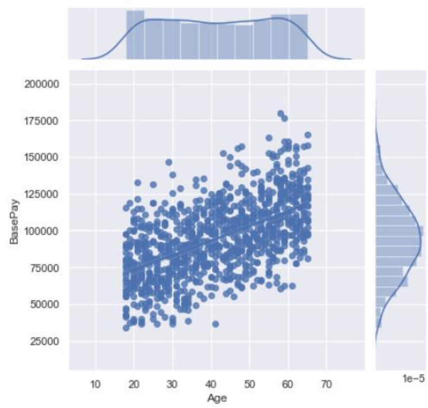


The difference between male and female is difficult to make.

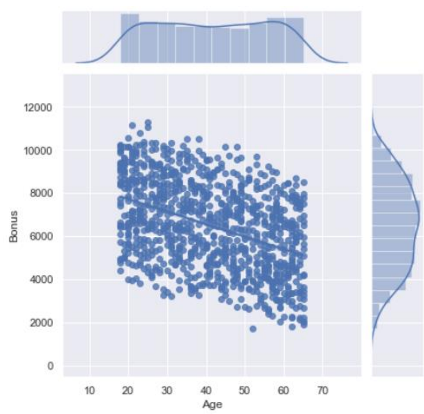
Finally, I decide to create a new target feature **TotalPay = BasePay + Bonus**.

FEATURES CORRELATION

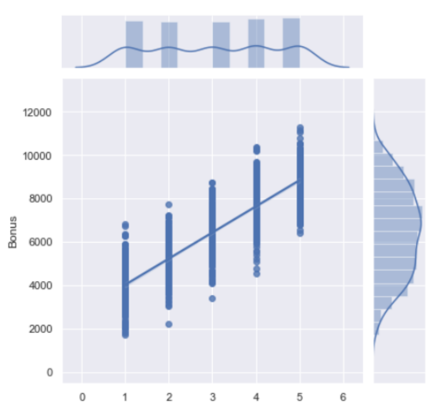
Base Pay is positively correlated with age:



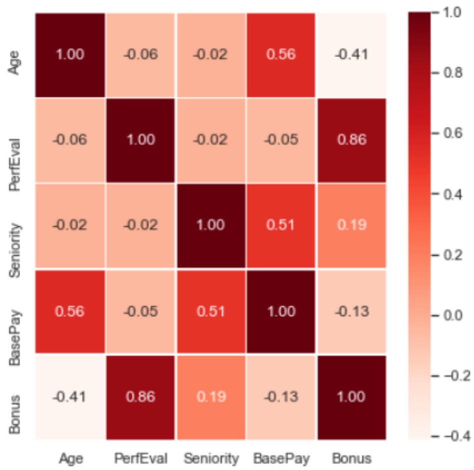
Where Bonus is negatively correlated with age:



The bonus is also correlated positively with the performance:



We also retrieved these key findings in the correlation matrix:



With:

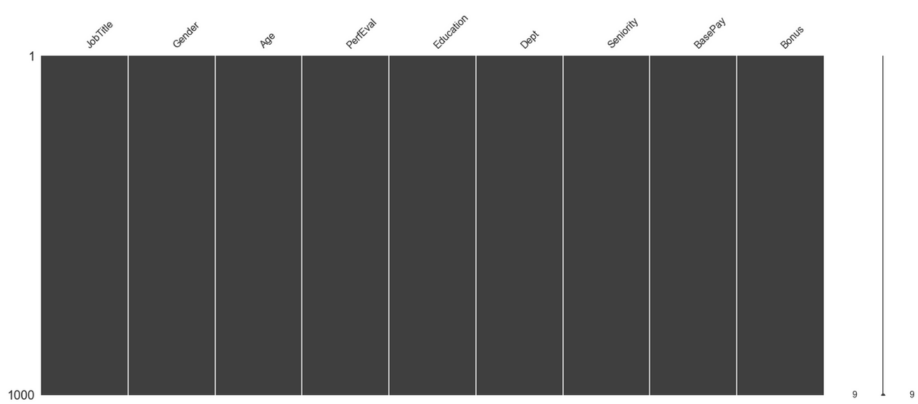
- Performance and Bonus are strongly correlated (0,86)
- Age is significatively correlated with
    - BasePay (0,56)
    - Age (-0,41)

➔ As we have correlated data, we will probably need later to select the features we keep for models evaluation.

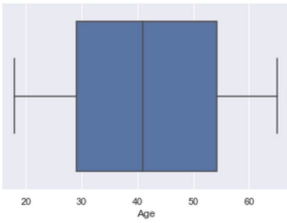
MISSING VALUES

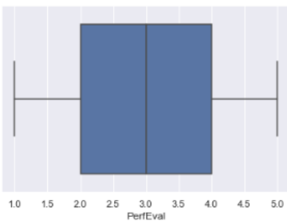
In this dataset, there are no missing values.

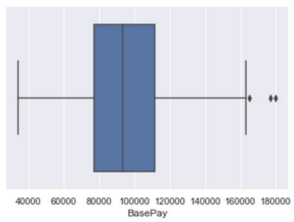
Indeed, there are no blank in the following missing data chart:

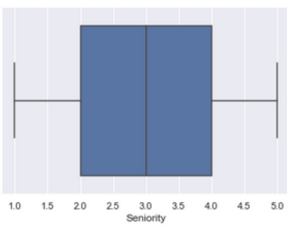


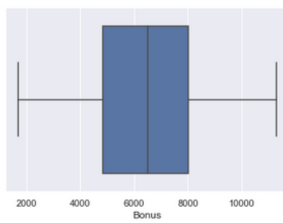
OUTLIERS











There are not a lot of outliers.

At this stage, I keep the outliers.

FEATURE ENGINEERING

Creation of new target feature **TotalPay = BasePay + Bonus**.

SCALING

Usually, numerical data of a dataset have different scales.

This is the case here where features **Age** have a bigger scales than other features:

	mean	std	min	25%	50%	75%	max
Age	41.393000	14.294856	18.000000	29.000000	41.000000	54.250000	65.000000
PerfEval	3.037000	1.423959	1.000000	2.000000	3.000000	4.000000	5.000000
Seniority	2.971000	1.395029	1.000000	2.000000	3.000000	4.000000	5.000000

Later, I will probably evaluate models that require scaled features. So, I will need to proceed to numerical data scaling.

ENCODING

In the dataset, there are categorical data and ordinal data. They need to be encoded before they can be included in certain models.

Categorical data overview before encoding:

	count	unique	top	freq
JobTitle	1000	10	Marketing Associate	118
Gender	1000	2	Male	532
Education	1000	4	High School	265
Dept	1000	5	Operations	210

Ex with the first 5 lines of the dataset:

	JobTitle	Gender	Education	Dept
0	Graphic Designer	Female	College	Operations
1	Software Engineer	Male	College	Management
2	Warehouse Associate	Female	PhD	Administration
3	Software Engineer	Male	Masters	Sales
4	Graphic Designer	Male	Masters	Engineering

➔ Education feature will become ordinal

From

Education
College
College
PhD
Masters
Masters

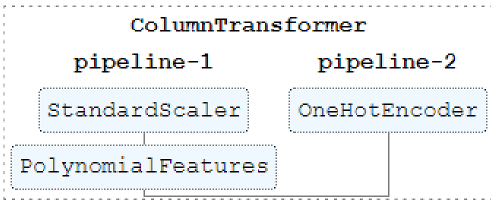
to

Education
1
1
3
2
2

➔ Other categorical features will be one hot encoded

The preprocessing pipeline looks like this:

- Pipeline 1 for numerical features
- Pipeline 2 for categorical feature



TRAINING REGRESSION MODELS

➔ Summary of training at least three linear regression models which should be variations that cover using a simple linear regression as a baseline, adding polynomial effects, and using a regularization regression. Preferably, all use the same training and test splits, or the same cross-validation method.

4 linear regression models are evaluated:

- Linear Regression
- Lasso Regression
- Ridge Regression
- ElasticNet Regression

1) First evaluation of 4 models

They are first evaluated with no parameter and with polynomial effects degree 8.

Results are the following with:

- metric R2 on train
- metric R2 on test
- metric RMSE on train
- metric RMSE on test

	Linear	Lasso	Ridge	Elastic
0	9.251588e-01	9.179443e-01	9.215593e-01	7.355251e-01
1	3.482169e-01	6.807452e-01	4.689385e-01	5.370317e-01
2	4.617355e+07	5.062455e+07	4.839424e+07	1.631687e+08
3	4.325115e+08	2.118517e+08	3.524028e+08	3.072174e+08

Linear, Lasso and Ridge have the same R2 around 0,92 on the train set.

➔ Only Lasso has a good R2 on the test set: 0,68 and the lowest error (RMSE)

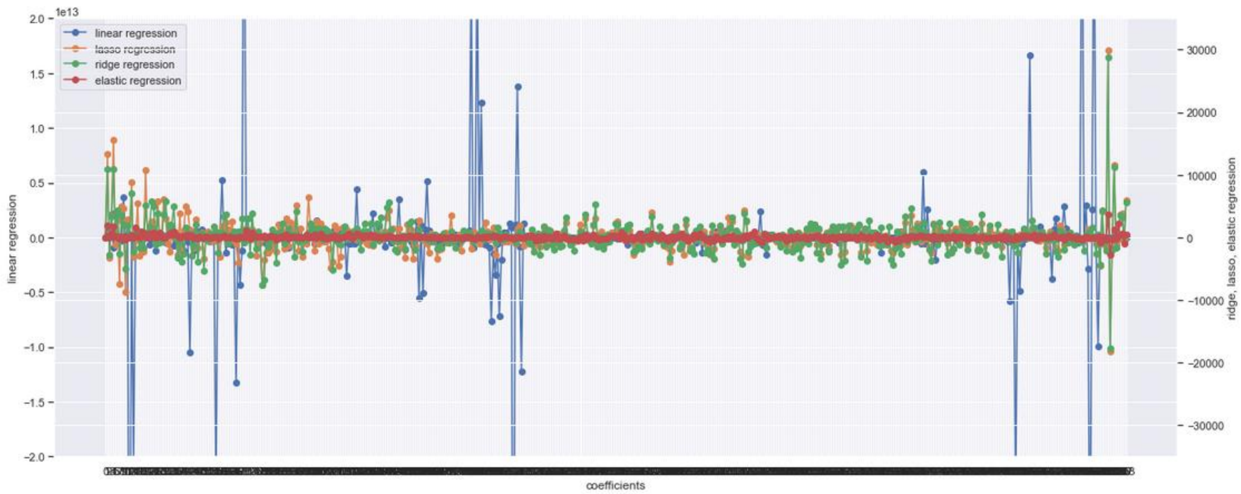
Here are coefficient statistics:

	Linear regression	Lasso regression	Ridge regression	Elastic regression
count	5.090000e+02	509.000000	509.000000	509.000000
mean	1.273110e+12	1431.141758	1494.968819	222.652829
std	5.155597e+12	2226.769491	2048.661926	306.198877
min	1.747561e-10	0.000000	0.000000	0.000000
25%	3.408399e+03	336.728565	402.213118	66.541142
50%	7.753652e+10	871.001942	961.788791	146.264830
75%	4.343730e+11	1785.153607	1990.894053	269.386659
max	4.715393e+13	29986.743054	28779.535410	3743.870085

Number of features after regularization:

```
Linear regression      509
Lasso regression       500
Ridge regression       508
Elastic regression     508
```

Let's have a look to the plot of the magnitude of coefficients obtained from these regressions:



➔ ElasticNet is good on regularization: near 0.

2) <u>Second evaluation of 4 models</u>

They are then evaluated with <u>parameters</u> (alpha and ratio) and with polynomial effects degree 8.

Results are the following with:

- metric R2 on train
- metric R2 on test
- metric RMSE on train
- metric RMSE on test

	Linear	Lasso	Ridge	Elastic
0	9.251588e-01	9.181274e-01	9.251593e-01	9.179023e-01
1	3.482169e-01	6.745970e-01	3.478641e-01	6.783835e-01
2	4.617355e+07	5.051159e+07	4.617325e+07	5.065047e+07
3	4.325115e+08	2.159315e+08	4.327456e+08	2.134189e+08

All regression have almost the same R2 around 0,92 on the train set.

➔ Only Lasso & ElasticNet have a good R2 on the test set, around 0,68, and the lowest error (RMSE)

Regression Analysis – Total reward

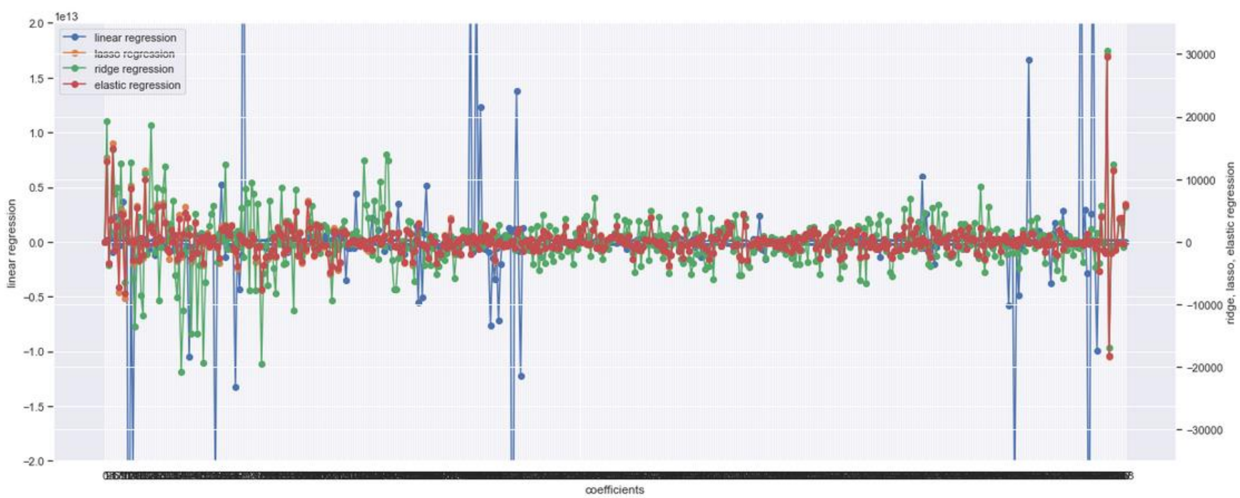
Here are coefficient statistics:

	Linear regression	Lasso regression	Ridge regression	Elastic regression
count	5.090000e+02	509.000000	509.000000	509.000000
mean	1.273110e+12	1525.676283	2848.807798	1443.846697
std	5.155597e+12	2256.663442	3522.785092	2167.858964
min	1.747561e-10	0.000000	0.000000	0.000000
25%	3.408399e+03	410.755787	719.309928	378.808738
50%	7.753652e+10	913.187132	1691.699951	890.741147
75%	4.343730e+11	1926.982313	3603.612156	1828.341063
max	4.715393e+13	29932.641005	30598.760549	29592.786828

Number of features after regularization:

```
Linear regression      509
Lasso regression       508
Ridge regression       508
Elastic regression     508
```

Let's have a look to the plot of the magnitude of coefficients obtained from these regressions:



➔ ElasticNet is good on regularization: near 0.

MODEL RECOMMENDATION

- ➔ A paragraph explaining which of your regressions you recommend as a final model that best fits your needs in terms of accuracy and explainability.

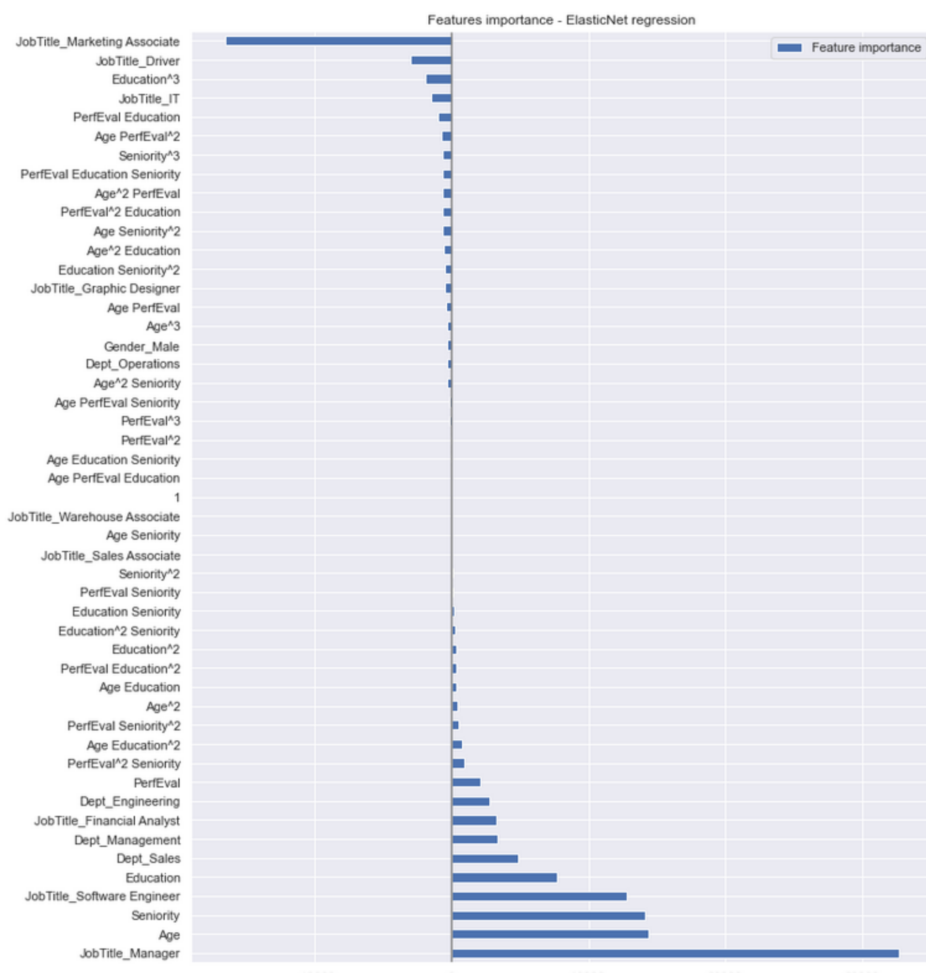
I recommend the ElasticNet regression model as it is the one of the lowest error (RMSE: Root Mean Square Error). It combines the regularization of both lasso and Ridge.

KEY FINDINGS AND INSIGHTS

- ➔ Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your linear regression model.

The models are run once again but this time with polynomial effects degree 3.

Resulting regression coefficients are then plot:



Job titles "Marketing associate" and "Drivers" drive lower total reward whereas job titles such as "Software engineer" and "Manager" drive higher total reward.

Education, Seniority and Age drive also higher total reward.

SUGGESTIONS FOR NEXT STEPS IN ANALYZING THIS DATA

➔ Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model adding specific data features to achieve a better explanation or a better prediction.

In this work, I have not managed correlated features. Correlated features induce instabilities in the coefficients of linear models and their effects cannot be well teased apart.

So, I would continue both the univariate and multivariate analysis.

I could also look at new features than could be created through feature engineering.

APPENDIX: CODE

The code for this project is available on GITHUB:

https://github.com/Olivier-FONTAINE/IBM-Machine-Learning-professional-certificate/blob/main/02-REG-Total%20Reward.ipynb