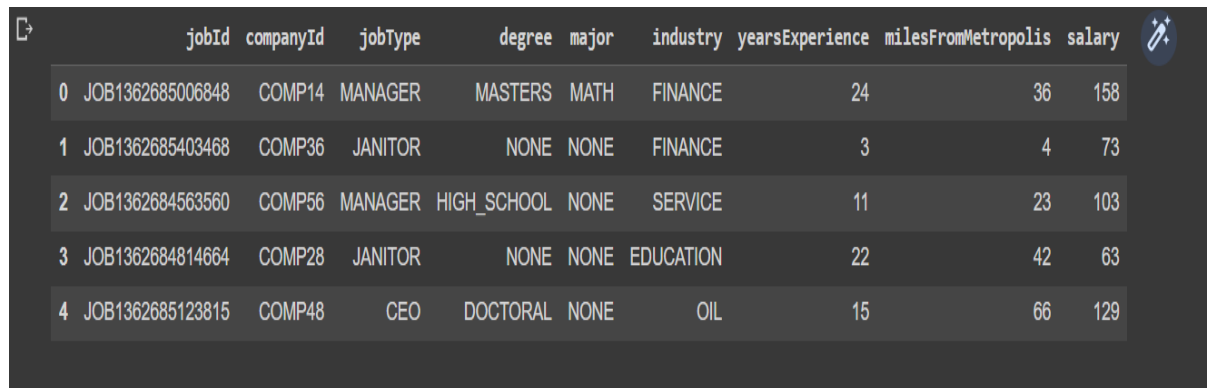


## Data Exploration & preparation:

- 1- Reading dataset: reading five columns from the data to discover Column's name and columns content



	jobId	companyId	jobType	degree	major	industry	yearsExperience	milesFromMetropolis	salary
0	JOB1362685006848	COMP14	MANAGER	MASTERS	MATH	FINANCE	24	36	158
1	JOB1362685403468	COMP36	JANITOR	NONE	NONE	FINANCE	3	4	73
2	JOB1362684563560	COMP56	MANAGER	HIGH_SCHOOL	NONE	SERVICE	11	23	103
3	JOB1362684814664	COMP28	JANITOR	NONE	NONE	EDUCATION	22	42	63
4	JOB1362685123815	COMP48	CEO	DOCTORAL	NONE	OIL	15	66	129

- 2- Data shape: (900000,9)

- 900000 Rows.
- 9 Columns.

- 3- Data describe

statistics values for each numeric column.

	yearsExperience	milesFromMetropolis	salary
count	900000.000000	900000.000000	900000.000000
mean	11.991183	49.525906	116.067520
std	7.211222	28.883348	38.717799
min	0.000000	0.000000	0.000000
25%	6.000000	25.000000	88.000000
50%	12.000000	50.000000	114.000000
75%	18.000000	75.000000	141.000000
max	24.000000	99.000000	301.000000

#### 4- Find duplicated

Check if data has duplicated values to drop them.

The Data has no duplicated values.

```
df.duplicated().sum()  
0
```

#### 5- Find null values for each column

There are no null values to process or delete them.

```
df.isnull().sum()  
jobId      0  
companyId  0  
jobType    0  
degree     0  
major      0  
industry   0  
yearsExperience  0  
milesFromMetropolis  0  
salary     0  
dtype: int64
```

6- Find info about the dataset and explore data types of data.

#	Column	Non-Null Count	Dtype
0	jobId	900000 non-null	object
1	companyId	900000 non-null	object
2	jobType	900000 non-null	object
3	degree	900000 non-null	object
4	major	900000 non-null	object
5	industry	900000 non-null	object
6	yearsExperience	900000 non-null	int64
7	milesFromMetropolis	900000 non-null	int64
8	salary	900000 non-null	int64

dtypes: int64(3), object(6)

Found three columns with (int64) data type [yearsExperience, milesFromMetropolis, salary].

Six columns with (object) data type [JobId, CompanyID, JopType, degree, major, industry].

Object columns should be processed Either using a label encoder or using one hot encoder.

Numeric columns should be scaled or normalized.

7- Remove unwanted chars from (job Id, company id) columns to convert from object to float data type.

When train models it's better to have the data with float data type.

Before Remove unwanted chars, jobId = JOB1362685006848

After Remove unwanted chars, jobId = 1362685006848

Before Remove unwanted chars, companyId = COMP14

After Remove unwanted chars, companyId = 14

## 8- show unique values for numeric columns.

Understanding unique values provides insights into the diversity and distribution of the data.

```
Column Name : yearsExperience
Column unique values : [24.  3. 11. 22. 15.  1. 12. 13. 17. 10.  0. 16.  5. 14.  2.  9. 18. 19.
 21. 23.  8.  6. 20.  4.  7.]

Column Name : milesFromMetropolis
Column unique values : [36.  4. 23. 42. 66. 51. 21. 96.  9. 74. 26.  0. 90. 61. 10. 71. 49. 41.
 13. 95. 68. 67. 69. 44. 79. 37. 53.  1. 40. 63. 14. 57.  2. 98. 18. 91.
 38.  8. 56. 16. 19. 54. 81. 48. 39. 15.  3. 62. 24. 12. 20.  5. 86. 45.
 77. 28. 34. 30. 11. 55. 27. 99. 80. 47. 75. 76. 46. 89. 22. 87. 70. 59.
 65. 52. 43. 25. 33.  7. 94. 73. 72. 31. 83. 35. 85. 88. 84. 29.  6. 60.
 97. 82. 58. 92. 64. 50. 32. 93. 78. 17.]
```

## 9- show unique values for categorical columns.

Understanding unique values provides insights into the diversity and distribution of the data.

```
Column Name : jobType
Column unique values : ['MANAGER' 'JANITOR' 'CEO' 'CTO' 'JUNIOR' 'CFO' 'VICE_PRESIDENT' 'SENIOR']
SENIOR          113266
VICE_PRESIDENT  112755
CTO             112648
JANITOR         112427
CEO             112357
MANAGER         112356
JUNIOR          112196
CFO             111995
Name: jobType, dtype: int64

Column Name : degree
Column unique values : ['MASTERS' 'NONE' 'HIGH_SCHOOL' 'DOCTORAL' 'BACHELORS']
HIGH_SCHOOL     213205
NONE            213095
BACHELORS       158006
DOCTORAL        157930
MASTERS         157764
Name: degree, dtype: int64
```

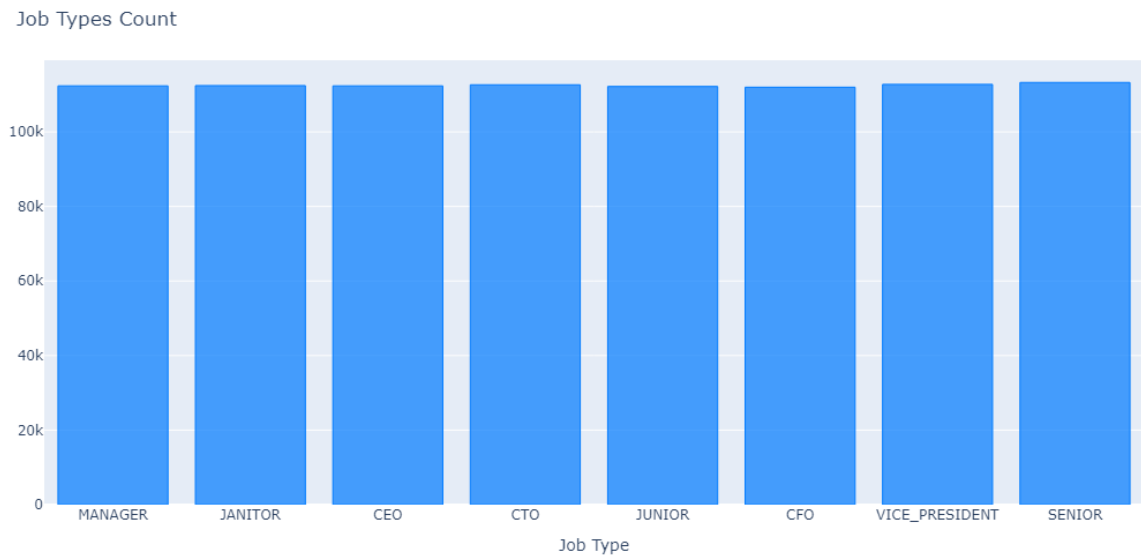
## 10- check for whitespaces in categorical columns.

leading or trailing whitespaces can create separate categories for the same value. Removing these whitespaces ensures proper encoding and avoids redundant or incorrect representations of categorical data.

In our data, there are no whitespaces or any chars that need to eliminate.

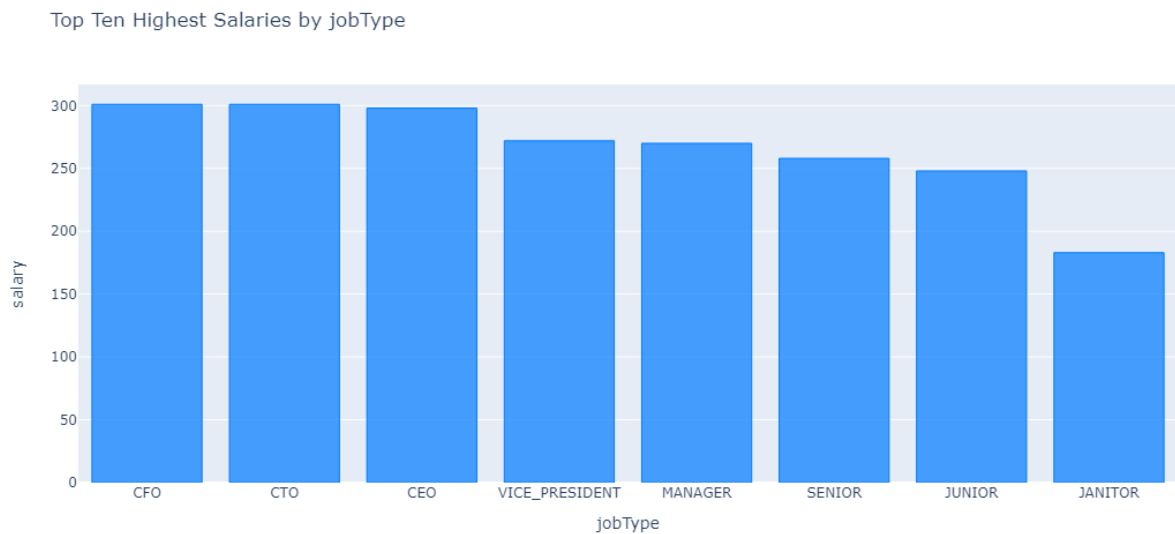
## Data Visualization:

### 1- job Type Column Visualization

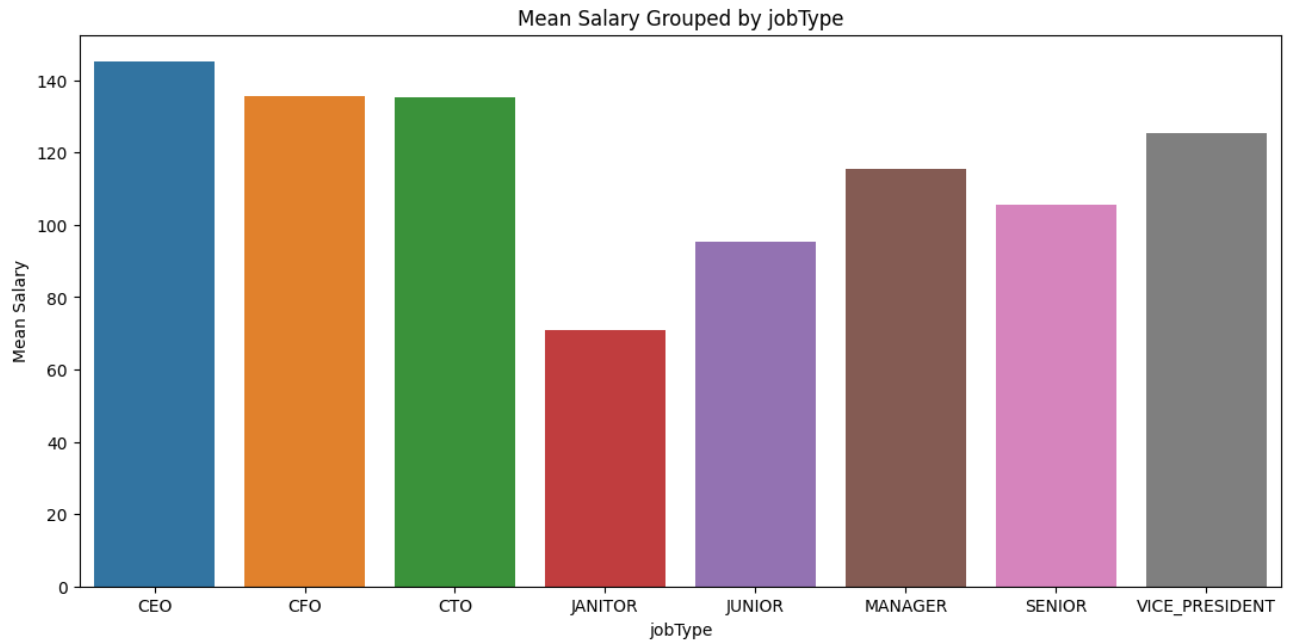


### 2- Plot The top Highest salaries according to Job Type.

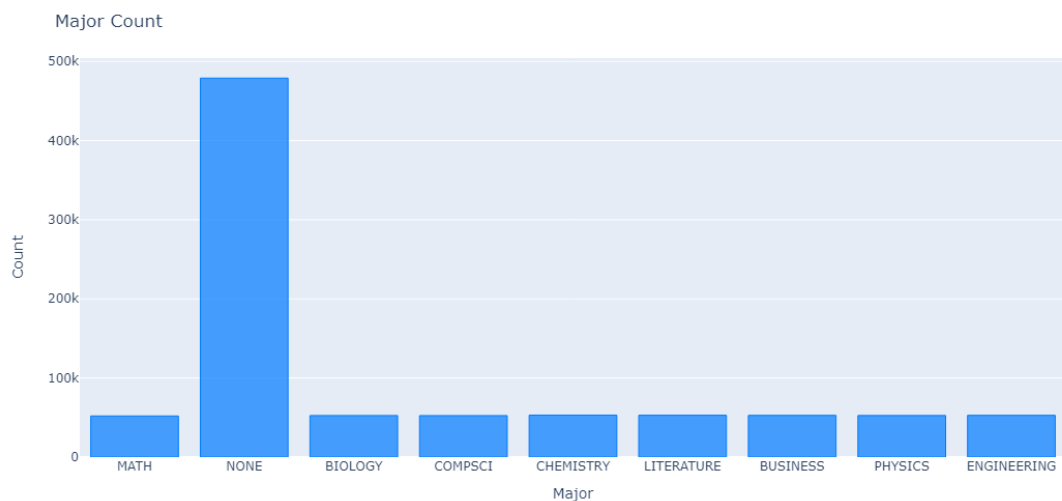
The CFO job type has the highest salary overall job types and the janitor has the lowest salary.



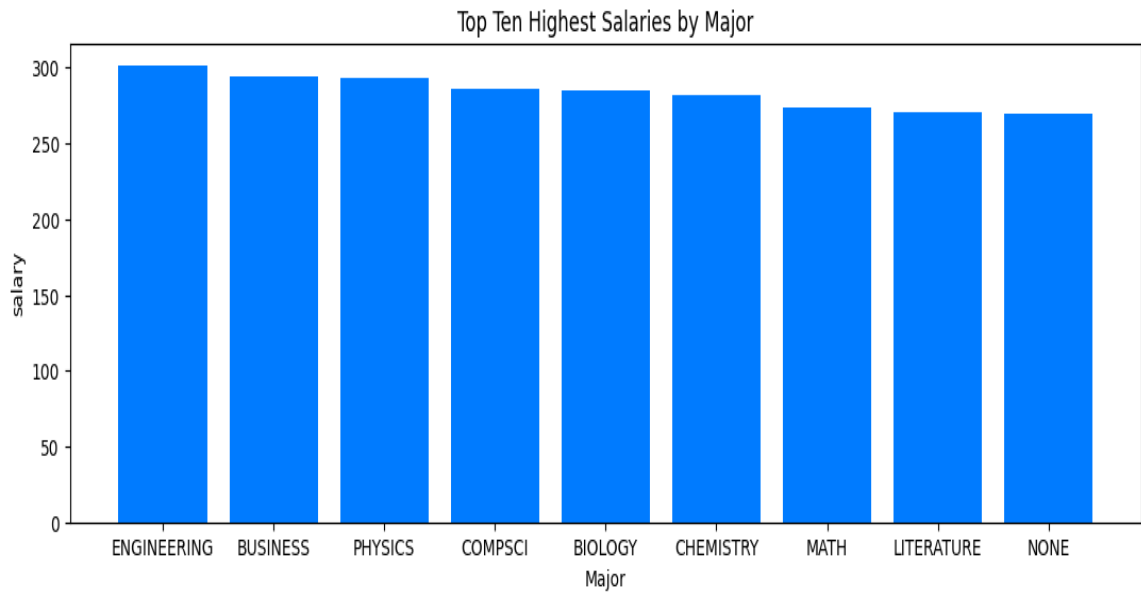
- 3- Group job Type values and calculate the mean salary.  
CEO has the highest mean salary, and the janitor has the lowest mean salary.



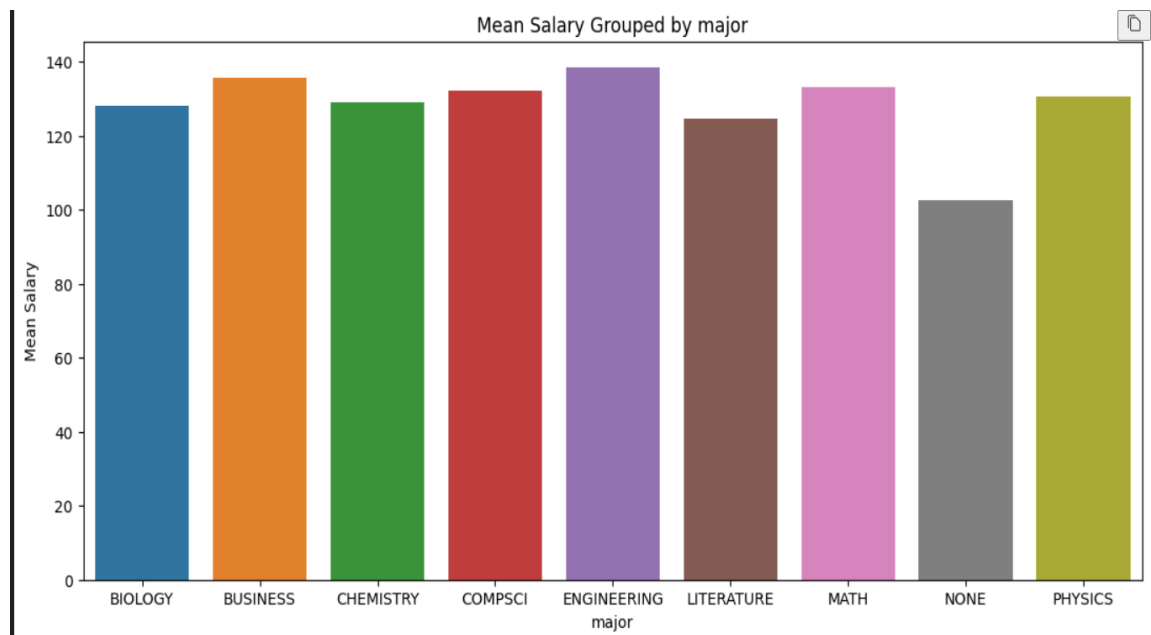
- 4- Major column visualization, Count plot.  
NONE major is the largest category among major categories



- 5- Top Ten Highest Salaries by Major.  
Engineering major has the highest salary value.

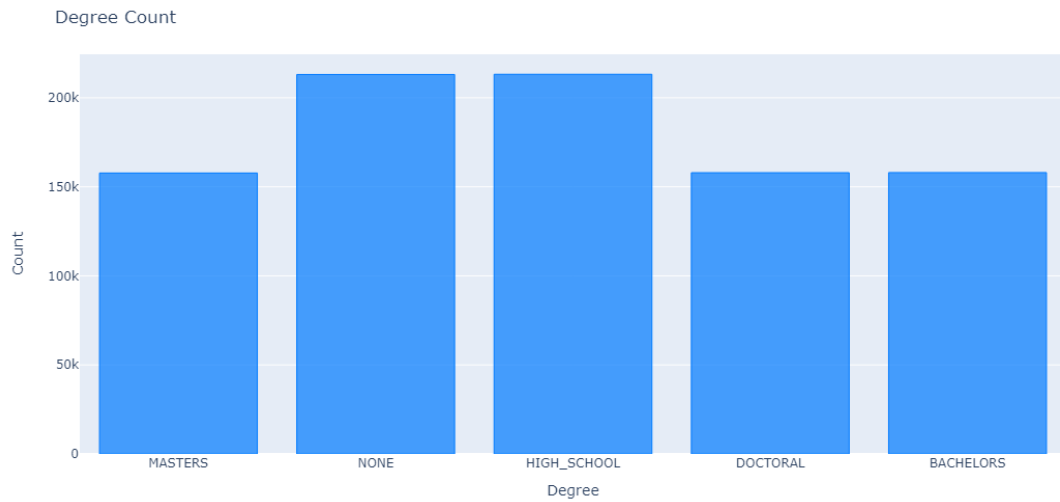


- 6- Group Major values and calculate the mean salary  
The Major that has the heights mean salary is Engineering major.



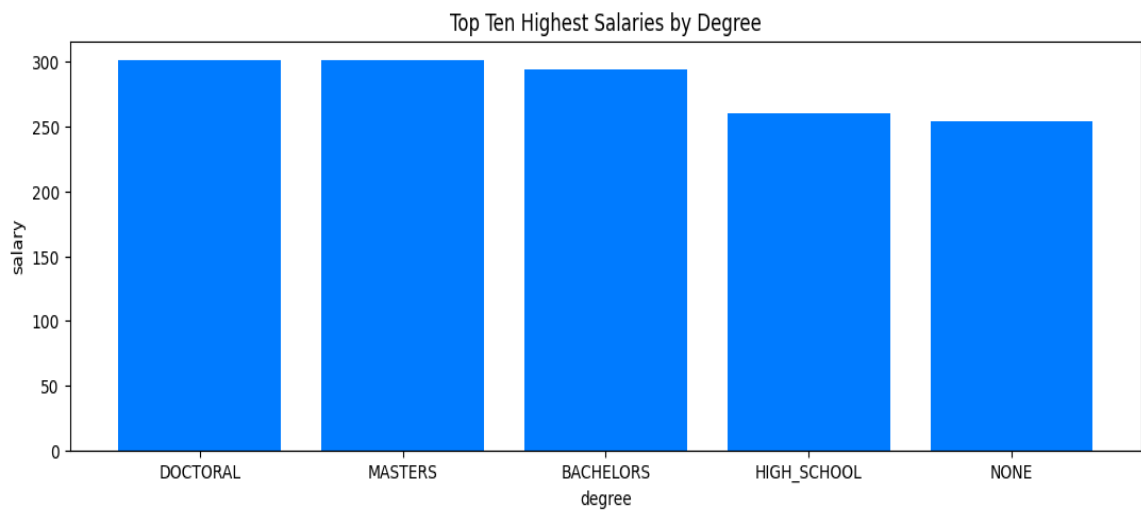
7- Degree column visualization, count plot

High school degree and NONE degree are the largest category over all Degree categories



8- Top Ten Highest Salaries by Degree

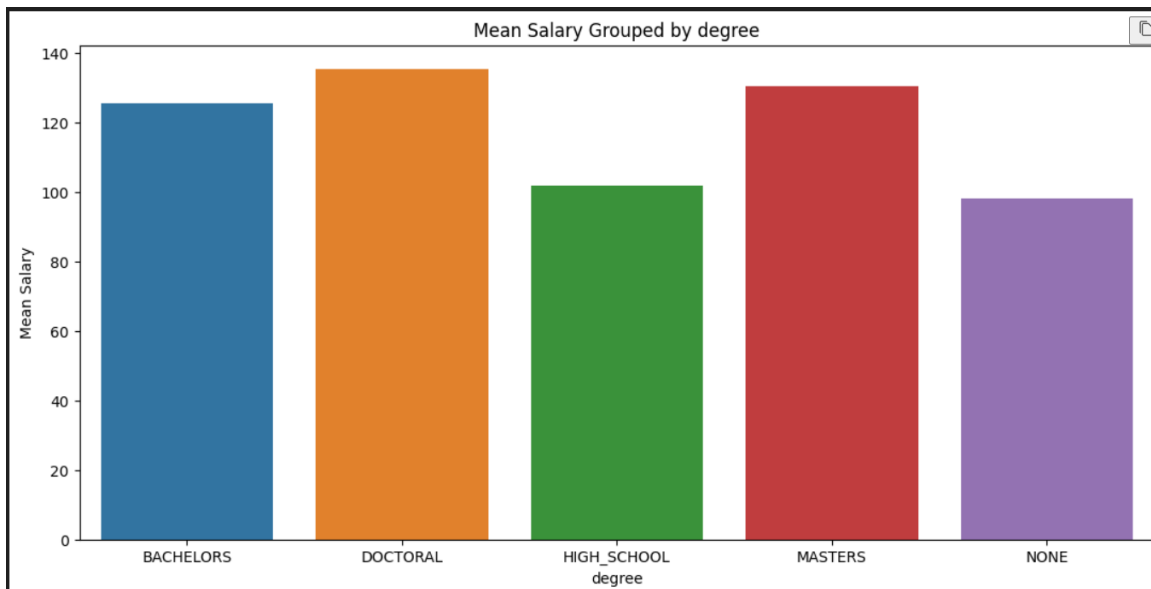
The highest salaries were for Doctoral and Master degree



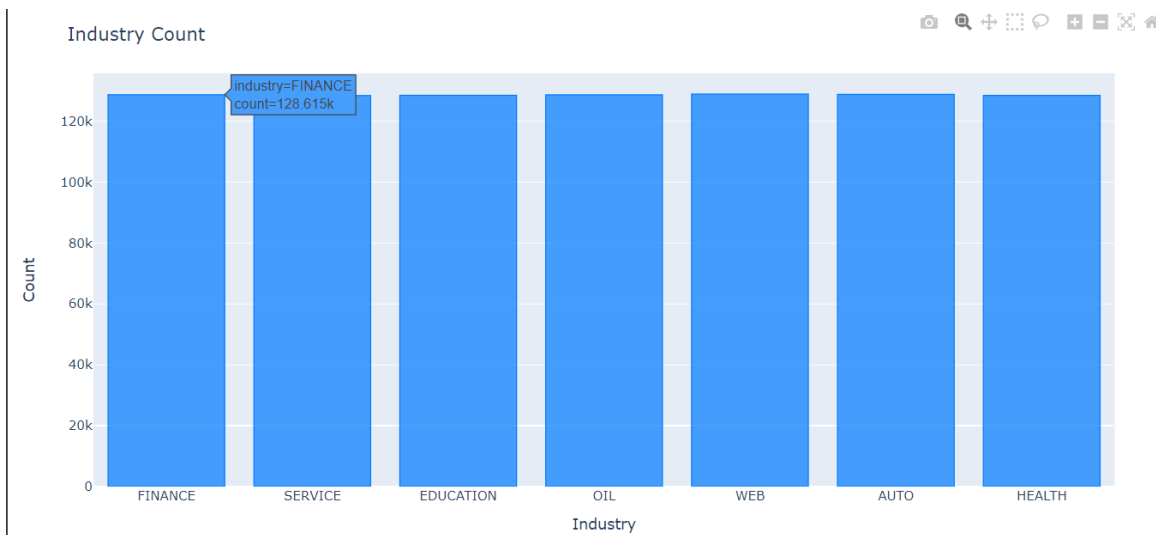


9- Group Degree values and calculate the mean salary

The degree has the highest mean salary is Doctoral and the lowest mean salary for None degree.

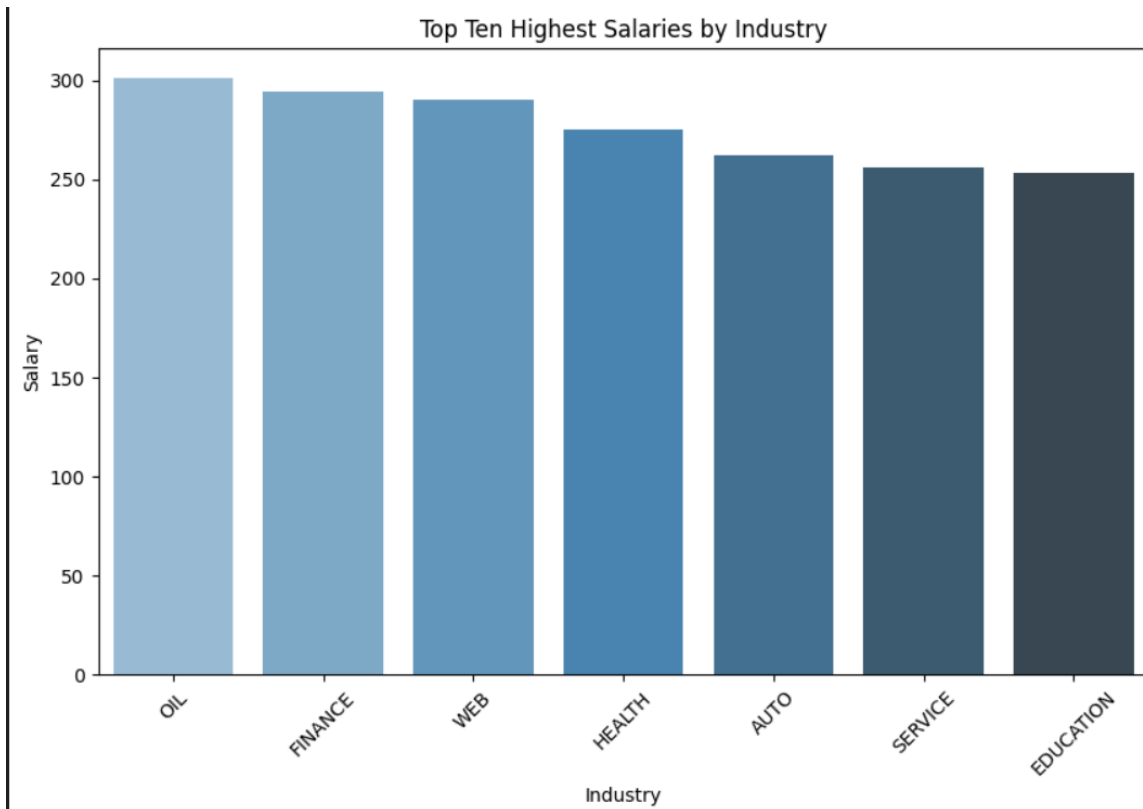


10- Industry column visualization, count plot.



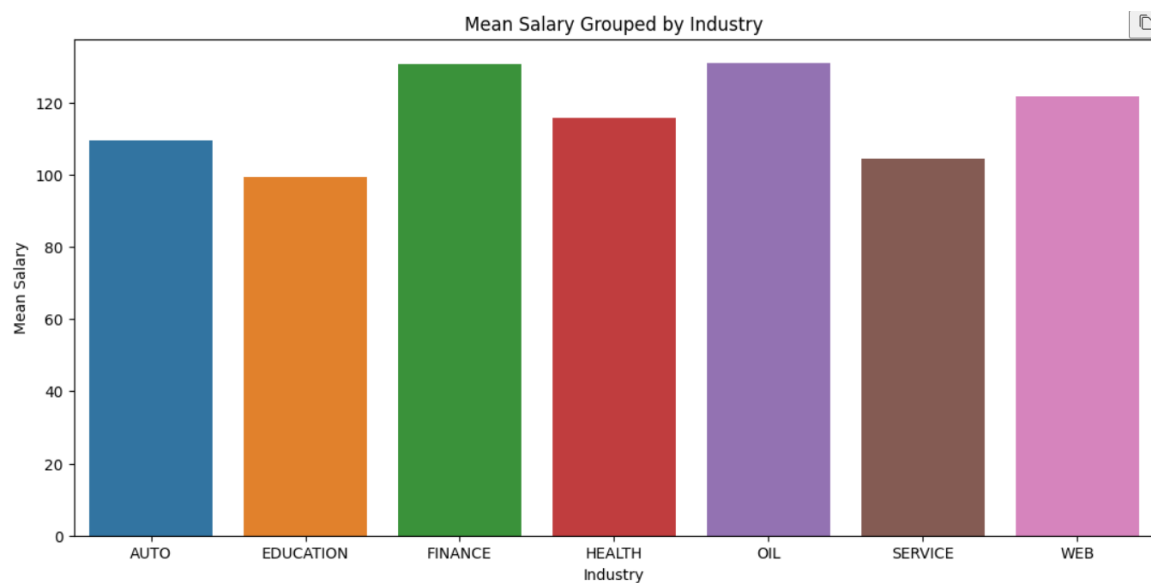
### 11- Top Ten Highest Salaries by Industry.

The highest salary for the oil industry and the lowest for the Education industry.



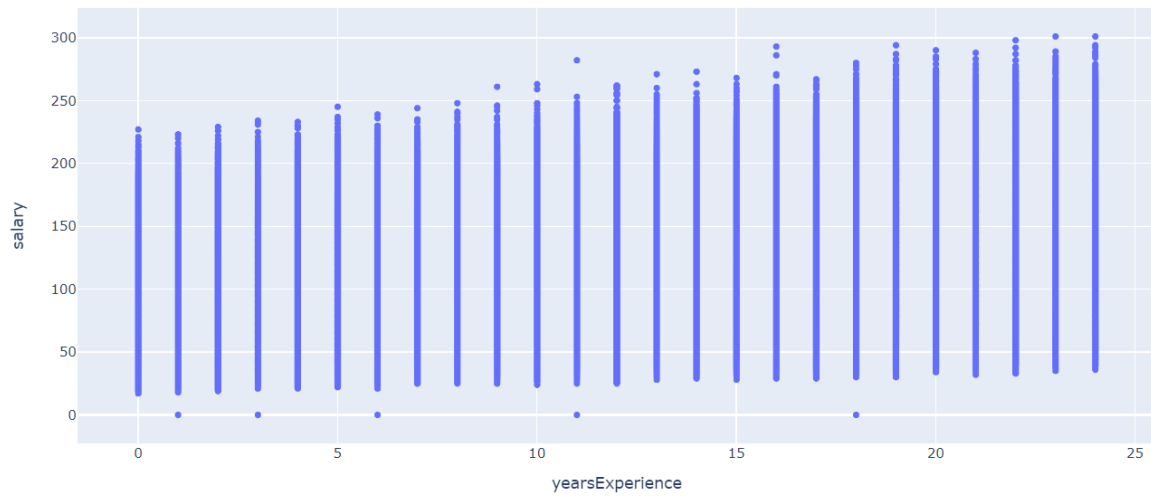
### 12- Group Industry values and calculate the mean salary.

Highest mean salaries for the oil and Finance industries.



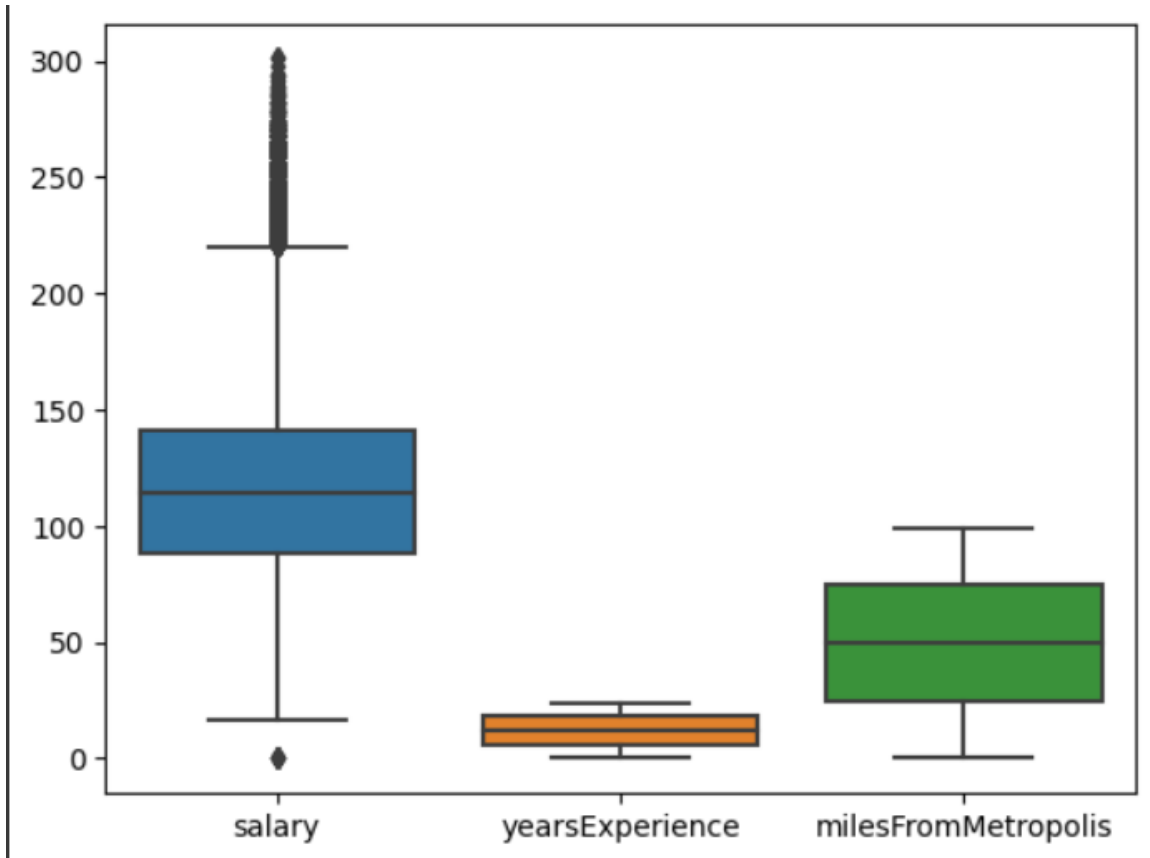
### 13- Year Experience column visualization, scatter plot between salary and Year Experience

The more years of experience, the higher the salary



14- Box plot to detect outliers in numeric columns.

The salary column in this dataset has both low and high outliers.



- We should deal with these outliers using the IQR method or Z-score method.

outliers can have a big impact on statistical analysis and machine learning because they impact calculations like mean and standard deviation, and they can skew hypothesis tests.

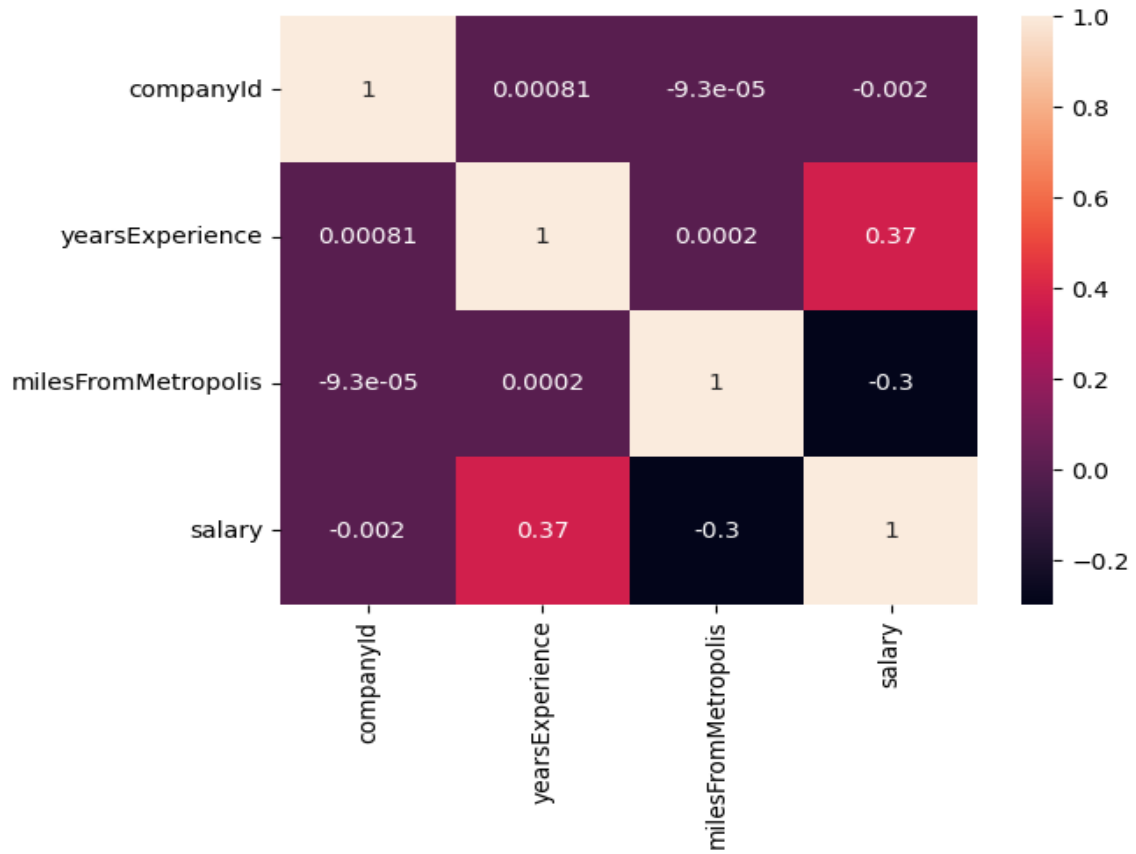
### 15- Mean Salary Grouped by Years of Experience

Mean salary increases as the years of experience increase.

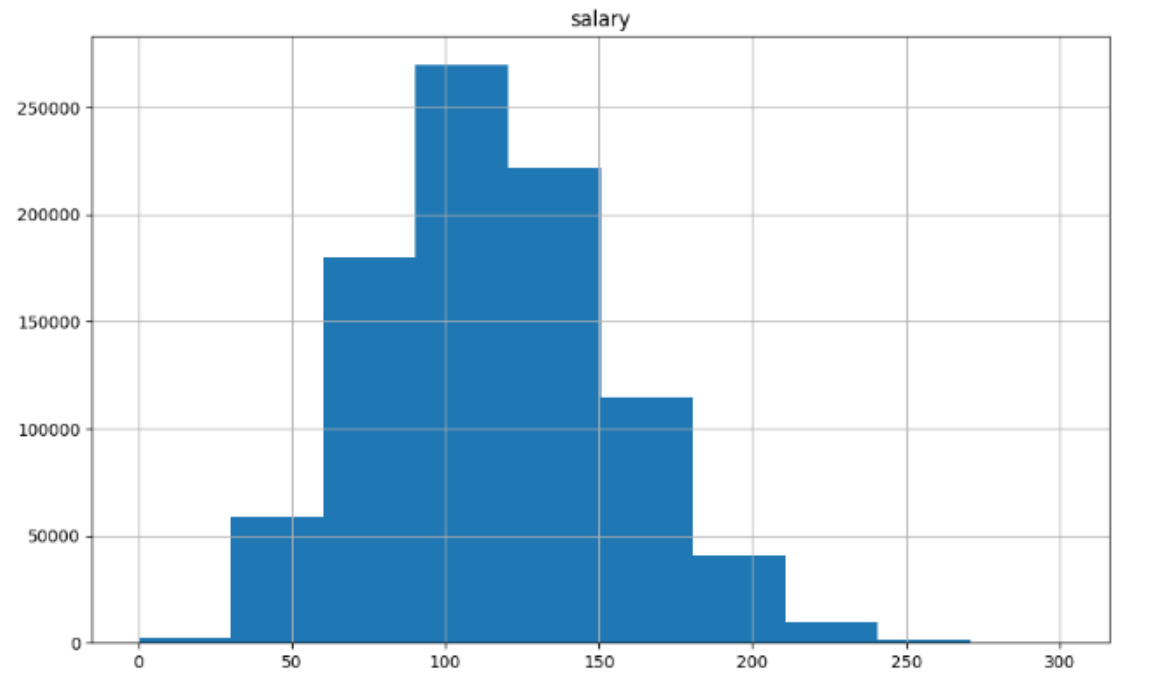


16- correlation Heatmap to show the correlation between columns.

salary and years of experience hast the most correlation value.



17- plot the Salary column histogram chart to see the distributed values.  
For this column, we should scale the values, because the accuracy of the model is more accurate when numeric columns be scaled.



## Modeling:

- 1- First drop the Job Id column because this column has no effect on the salary prediction and has no correlation with any other column.
- 2- Chose the Salary column to be the target column for prediction.
- 3- Preprocessing for numerical data: Standard scaler for the numeric feature because the machine learning models do better when we scale the values.
- 4- Preprocessing for categorical data: I used One Hot Encoder for the categorical feature because ml models can't deal with string values so we need to convert these strings to numerical numbers.
- 5- Chose machine learning models (XGBoost, LightGBM, Linear Regression) to train.
- 6- Split data for train and test sets with random state 42 and I chose the percent of the split to be 20%
- 7- Scale the Salary column to make the prediction model more accurate.
- 8- I used the pipeline on machine learning models to fast the process and made the code more clearly.
- 9- Chose MSE (Mean Squared Error) as evaluation metric.
- 10- Train the three models and chose the best model that has the lowest MSE value.



11- build a deep learning model to do more experiments to get the best result.

- Preprocessing for numerical data: Standard scaler for the numeric feature because the machine learning models do better when we scale the values.
- Preprocessing for categorical data: I used One Hot Encoder for the categorical feature because ml models can't deal with string values so we need to convert these strings to numerical numbers.
- Chose optimizer: ADAM, because it is the most used these days and its results are good.
- Train on 60 epochs, because the loss value stop changing after 60.
- Loss: mean squared error.

### comparison between models' results:

Model name	XGBoost	LightGBM	Linear Regression	Deep learning
MSE value	0.003960546	0.003944419	0.004241850	0.00403243

The comparison of the results of applying different models on the data with the given mean squared error (MSE) values suggests the following:

**XGBoost:** The XGBoost model achieved an MSE of 0.003960546. XGBoost is an ensemble learning method that uses gradient boosting to create a powerful predictive model. It is known for its ability to handle complex relationships in the data and deliver high predictive accuracy. With a low MSE value, the XGBoost model has performed well in capturing the patterns and predicting the target variable in the dataset.

**LightGBM:** The LightGBM model achieved an MSE of 0.003944419. LightGBM is another gradient-boosting framework that focuses on efficiency and speed. It is designed to handle large-scale datasets efficiently and can be a good choice for datasets with a large number of features. With a slightly lower MSE value compared to XGBoost, the LightGBM model has also performed well in capturing the data patterns and making accurate predictions.

**Linear Regression:** The Linear Regression model achieved an MSE of 0.004241850. Linear Regression is a basic statistical modeling technique that assumes a linear relationship between the features and the target variable. It is a simple and interpretable model, but it may not capture complex nonlinear relationships in the data as effectively as ensemble models like XGBoost and LightGBM. With a slightly higher MSE value, the Linear Regression model has performed comparatively less well in capturing the data patterns and predicting the target variable accurately.

Based on the MSE values, both XGBoost and LightGBM have shown better performance than Linear Regression in terms of capturing the data patterns and making accurate predictions. The lower MSE values indicate that XGBoost and LightGBM have better predictive capabilities for this specific dataset.

For now, the best model is LightGBM with MSE: 0.003944419597775922

**Note:** the best model to fit the data among the given options would be either XGBoost or LightGBM. It is recommended to further evaluate and compare these models based on other evaluation metrics, such as R-squared, cross-validation performance, and business-specific requirements, to make a final decision on the best model for your specific needs.

In prediction function:

Choose the best model (LIGHTGBM model) to predict

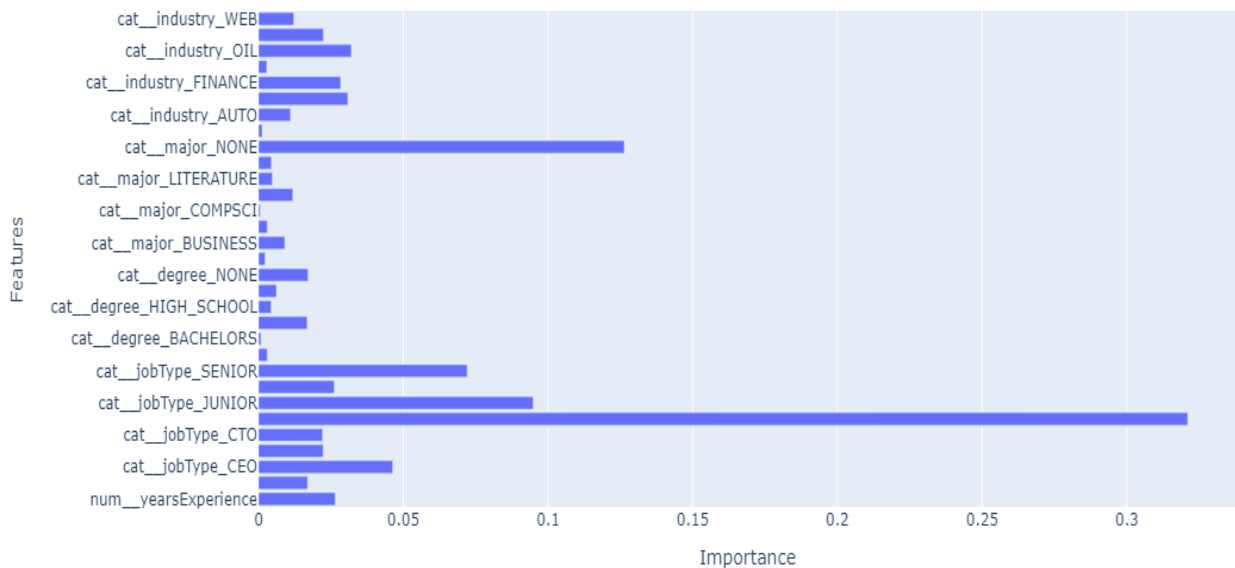
Use (standarScaler.inverse\_transform) to get the prediction value for the salary to the new data passed to the function

## Key findings:

- Feature importance is a valuable aspect of the XGBoost model that helps understand the relative significance of each feature in predicting the target variable. The feature importance of XGBoost can be obtained from the trained model and provides insights into which features have the most impact on the predictions.

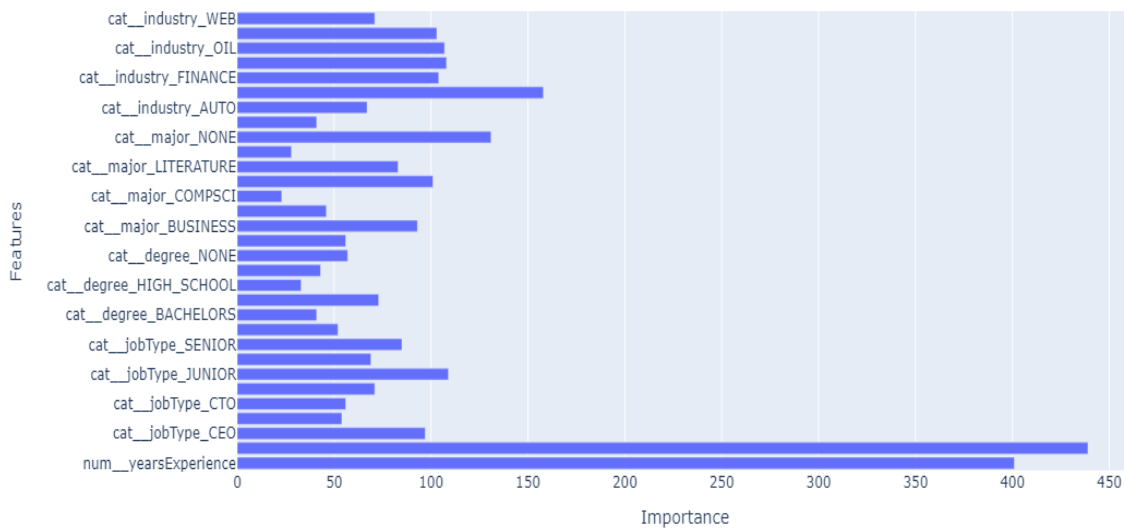
1- Based on the feature importance analysis of the XGBoost model, it can be concluded that the "job type" feature has the most impact on the prediction. The XGBoost model has determined that "job type" is the most influential feature in predicting the target variable and It implies that different job types might have distinct salary ranges and can significantly impact salary predictions.

Feature Importances - XGBoost Model



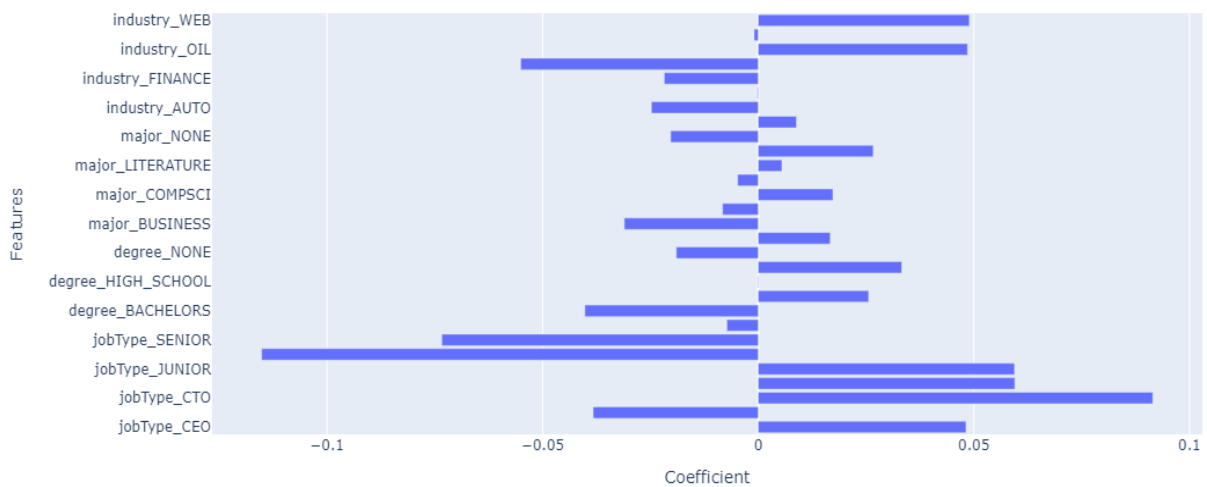
2- when using the LightGBM model for salary prediction, it is crucial to consider both "years of experience" and "job type" as the most influential factors for accurate and reliable predictions. Incorporating these features into the prediction model and understanding their impact will contribute to better predictions and deeper insights into salary patterns.

Feature Importances - LightGBM Model

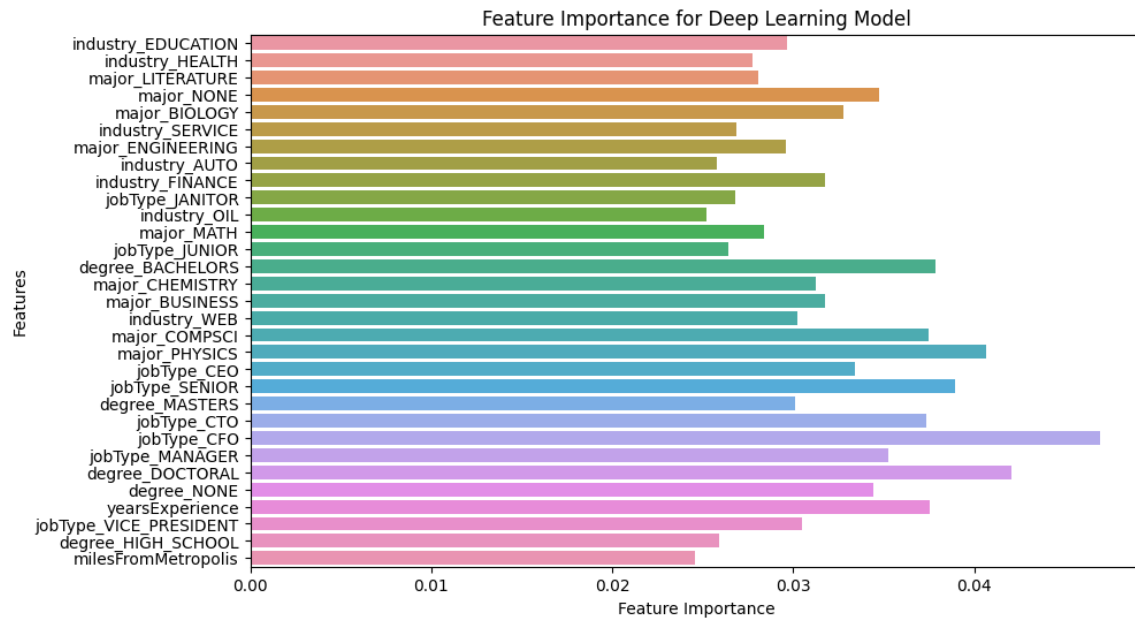


3- Considering the high coefficient value associated with the "job type" feature, it is important to emphasize the influence of this feature when making predictions or analyzing the dataset using [the Linear Regression model](#). The coefficient indicates the direction and magnitude of the relationship between the "job type" feature and the target variable (salary). A higher coefficient suggests a stronger impact on the predicted salary.

Feature Importances - Linear Regression Model



- 4- In the deep learning model, we found that the job type and the presence of a degree are indeed important features. Through the training process, the model learned to assign significant weights to these features when making predictions.



Finally, based on the feature importance analysis from all the models, it can be observed that "job type" has a significant impact on salary prediction, followed by "years of experience "

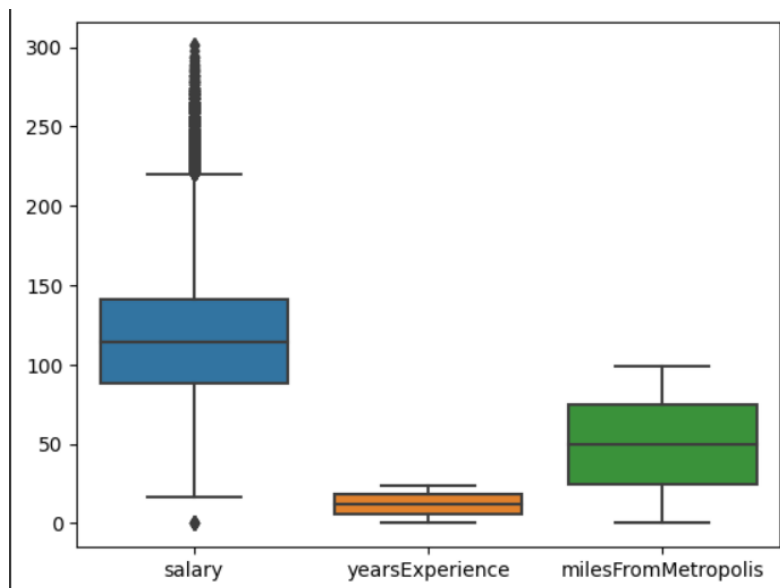
## Next Step:

suggestions for improving the model accuracy:

- 1- Outlier Handling: Analyze and handle outliers in the dataset. Outliers can significantly impact model performance and predictions. Consider applying appropriate techniques such as removing outliers, transforming them, or using robust models that are less sensitive to outliers.

In this data set, we have an outlier in the [salary column](#) and that was evident when the plot box plot.

I choose to just scale the values in the salary column and I don't process the outliers.



- 2- Hyperparameter Tuning: Optimize the hyperparameters of models. Fine-tuning the hyperparameters can significantly impact model performance, and search for pre-trained models for fine-tuning.



- 3- Collect More Data: If feasible, collect more data to increase the size of the dataset. A larger dataset can provide more diverse examples and help the model learn better patterns and improve its accuracy.
- 4- Add more features to allow the use of feature engineering to extract more useful data.  
like age, full-time or part-time job, gender...
- 5- In our training models ([XGBoost](#), [LightGBM](#), [Linear Regression](#), [Deep learning](#)) the difference in errors among the models is relatively small, so further analysis and consideration of other factors such as model interpretability, computational efficiency, and scalability may be necessary to make a comprehensive comparison and decision on model selection.

Final note:

I tried to train more models like Support Vector Regression (SVR) and RandomForestRegressor, but the time of training was so long and didn't allow me to continue training to get more accurate results.