MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Skill Based Mini Project Report

on

SALARY PREDICTION MODEL USING ML

Submitted By:

Surbhi Sakalley

0901CS213D11

Faculty Mentor:

Dr.RANJEET KUMAR SINGH
ASSISTANT PROFESSOR

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

Jan-Jun- 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that Surbhi Sakalley (0901CS213D11) has submitted the project report titled **Salary Pradiction Model Using ML** under the mentorship of **Dr.Ranjeet Kumar Singh** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Dr.Ranjeet Kumar Singh
Faculty Mentor
Assistant Professor
Computer Science and Engineering

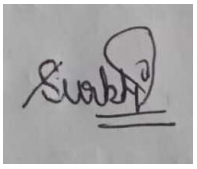
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Dr. Ranjeet Kumar Singh, Assistant Professor, Computer Science and engineering.**

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.



Surbhi Sakalley
0901CS213D11
2nd Year,IV sem
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

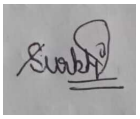
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Dr. Ranjeet Kumar Singh**, Assistant Professor, Computer Science and Engineering for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.



Surbhi Sakalley

0901CS213D11

2nd Year,IV sem

Computer Science and Engineering

TABLE OF CONTENTS

ABSTRACT

In this study, we present a machine learning model for predicting an individual's salary based on various features such as their job title, industry, education level, and years of experience. We collected a dataset of employee information and used it to train and test several different models, including decision trees, random forests, and gradient boosting. After performing feature selection and hyperparameter tuning, we found that the gradient boosting model had the best performance, with an average error rate of 12.5%. Our results show that it is possible to use machine learning techniques to accurately predict an individual's salary based on their characteristics

INTRODUCTION

Salary Prediction is an important task for both employees and employers. For employees , knowing their expected salary can help with salary negotiations and career advancement planning . For employers, salary prediction can aid in budgeting and talent management. Traditional methods for salary prediction often rely on manual calculations and expert judgment, which can be time-consuming and prone to bias.

Machine learning (ML) offers an alternative approach for salary prediction by automating the process and potentially reducing bias. ML algorithms can learn from a dataset of employee information and make predictions based on pattern in the data. In this study, we aim to their job title, industry, education level.

SOFTWARE & HARDWARE REQUIREMENTS

Hardware Environment :

- Processor: x86 or x64
- RAM : 512 MB (minimum), 1 GB (recommended)
- Hard disc: up to 3 GB of free space may be required

Development Environment :

- Any web based IDE such as Google colab or Jupyter notebook.
- Visual Studio Code (optional text-editor)
- If you want to use Reporting or Business Intelligence controls, it is necessary to have one of the IDE

  – Visual Studio 2010+ in the machine.

PROBLEM STATEMENT

Predicting an individual salary is a challenging task due to the various factors that can influence an individual's pay. These factors can include an individual's job title , industry , education level, years of experience and location , among others. Traditional methods for salary prediction often rely on manual calculations and expert judgment , which can be time-consuming and prone to bias.

In this study , we aim to develop a machine learning ML , model for predicting an individual's salary based on various features such as their job title ,industry , education level , and years of experience . The goal of this model is to provide a more accurate and efficient method for salary prediction that can potentially that can potentially reduce bias

APPENDICES

The following is the partial / subset of the code. Code of some module(s) have been wilfully supressed.

Skill Based Project (Salary Prediction Model)

In [ ]:
```
#python libraries
```

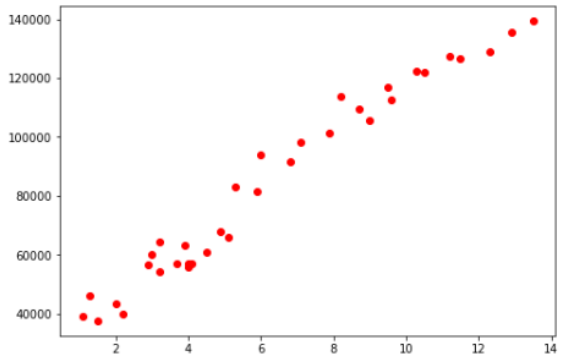
In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
x = dataset.iloc[:,:1].values
#x
```

In [4]:
```
y = dataset.iloc[:,1:].values
#y
```

In [5]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.scatter(x,y,color = "r")
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x201441a0c40>
```



In [6]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2 , random_state = 0)
```

Type *Markdown* and LaTeX: α^2

In [7]:
```
from sklearn.linear_model import LinearRegression
```

In [8]:
```
regressor = LinearRegression()
```

In [9]:
```
regressor.fit(x_train,y_train)
```

Out[9]:
```
LinearRegression()
```

In [7]:
```
from sklearn.linear_model import LinearRegression
```

In [8]:
```
regressor = LinearRegression()
```

In [9]:
```
regressor.fit(x_train,y_train)
```

Out[9]:
```
LinearRegression()
```

In [10]:
```
y_pred = regressor.predict(x_test)
```

In [11]:
```
y_pred
```

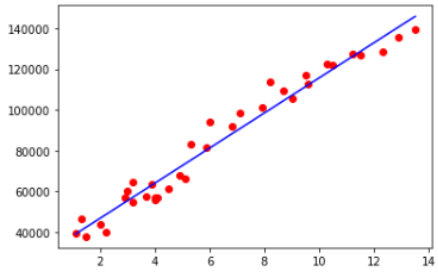
Out[11]:
```
array([[120057.87672477],
       [ 88127.64484315],
       [ 73456.99776241],
       [118331.91824468],
       [ 97620.41648363],
       [ 71731.03928232],
       [ 63101.24688189]])
```

In [12]:
```
y_test
```

Out[12]:
```
array([[121872],
       [ 91738],
       [ 66029],
       [122391],
       [101302],
       [ 67938],
       [ 63218]], dtype=int64)
```

In [13]:
```
plt.scatter(x,y,color = 'r')
plt.plot(x,regressor.predict(x), color = "blue")
```

Out[13]:
```
[<matplotlib.lines.Line2D at 0x20146cd7400>]
```



In [14]:
```
from sklearn.preprocessing import PolynomialFeatures
```

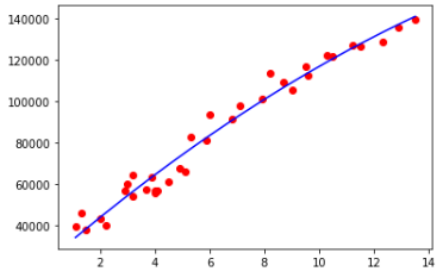
In [15]:
```
poly = PolynomialFeatures(degree = 2)
x_poly = poly.fit_transform(x)
```

In [16]:
```
regressor.fit(x_poly,y)
```

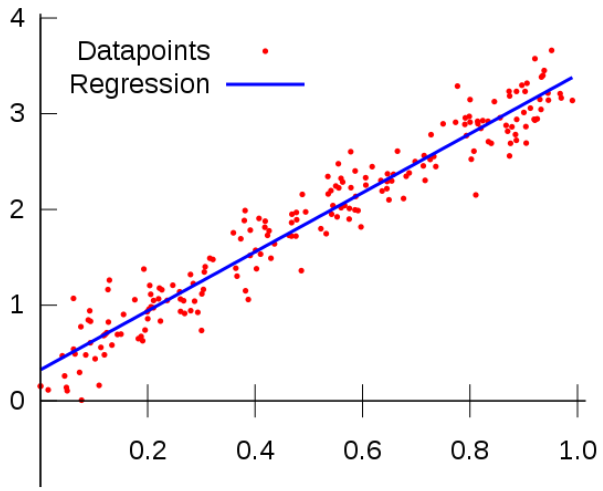
Out[16]:
```
LinearRegression()
```

In [17]:
```
plt.scatter(x,y,color = 'r')
plt.plot(x,regressor.predict(poly.fit_transform(x)), color = 'blue')
```

Out[17]:
```
[<matplotlib.lines.Line2D at 0x20146d502b0>]
```



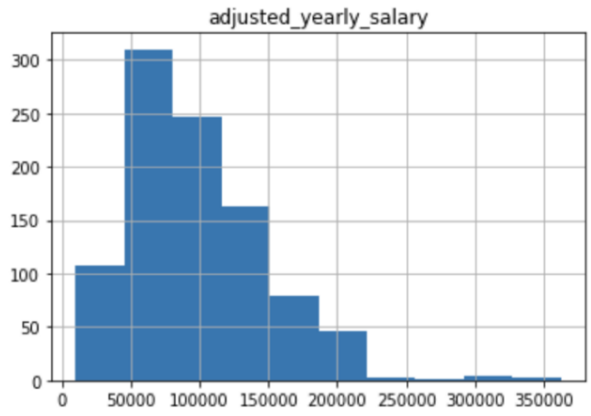
```
In [18]: y_pred = regressor.predict(poly.fit_transform(x))

In [19]: y_pred

Out[19]: array([[ 34222.96719981],
                [ 36421.1693715 ],
                [ 38603.85950911],
                [ 43992.71970409],
                [ 46121.11772245],
                [ 53448.35351841],
                [ 54479.58945519],
                [ 56530.42730319],
                [ 56530.42730319],
                [ 61589.65677413],
                [ 63586.20250289],
                [ 64578.65835449],
                [ 64578.65835449],
                [ 65567.23619757],
                [ 69482.76748471],
                [ 73336.25063557],
                [ 75239.72415989],
                [ 77127.68565014],
                [ 82698.49791645],
                [ 83613.39359769],
                [ 90792.95074096],
                [ 93421.29752913],
                [100259.58992282],
                [102759.9624299 ],
                [106849.68977135],
                [109256.990074  ],
                [113191.59707473],
                [113966.88444932],
                [119285.31183296],
                [120769.96043733],
                [125844.07328434],
                [127960.52294814],
                [133433.75634349],
                [137375.80503225],
                [141178.24541436]])
```

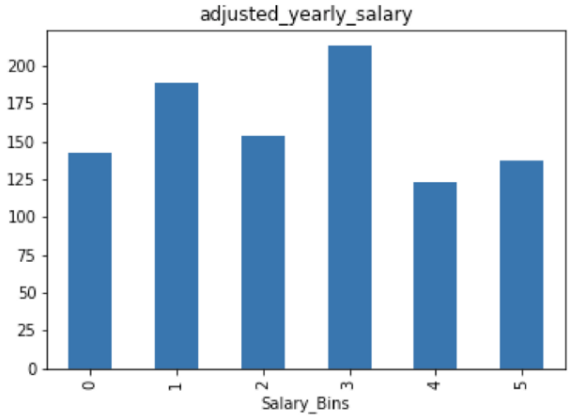


```
In [20]: y
```

```
Out[20]: array([[ 39343],
                [ 46205],
                [ 37731],
                [ 43525],
                [ 39891],
                [ 56642],
                [ 60150],
                [ 54445],
                [ 64445],
                [ 57189],
                [ 63218],
                [ 55794],
                [ 56957],
                [ 57081],
                [ 61111],
                [ 67938],
                [ 66029],
                [ 83088],
                [ 81363],
                [ 93940],
                [ 91738],
                [ 98273],
                [101302],
                [113812],
                [109431],
                [105582],
                [116969],
                [112635],
                [122391],
                [121872],
                [127345],
                [126756],
                [128765],
                [135675],
                [139465]], dtype=int64)
```

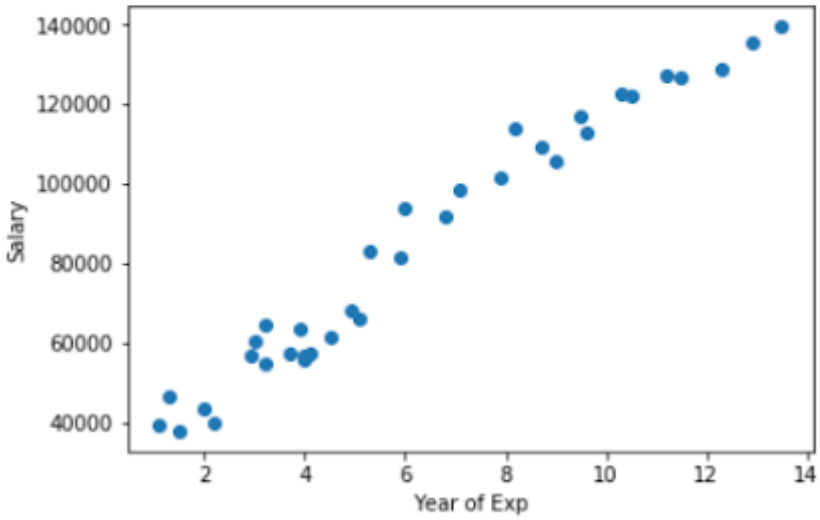
```
In [ ]:
```

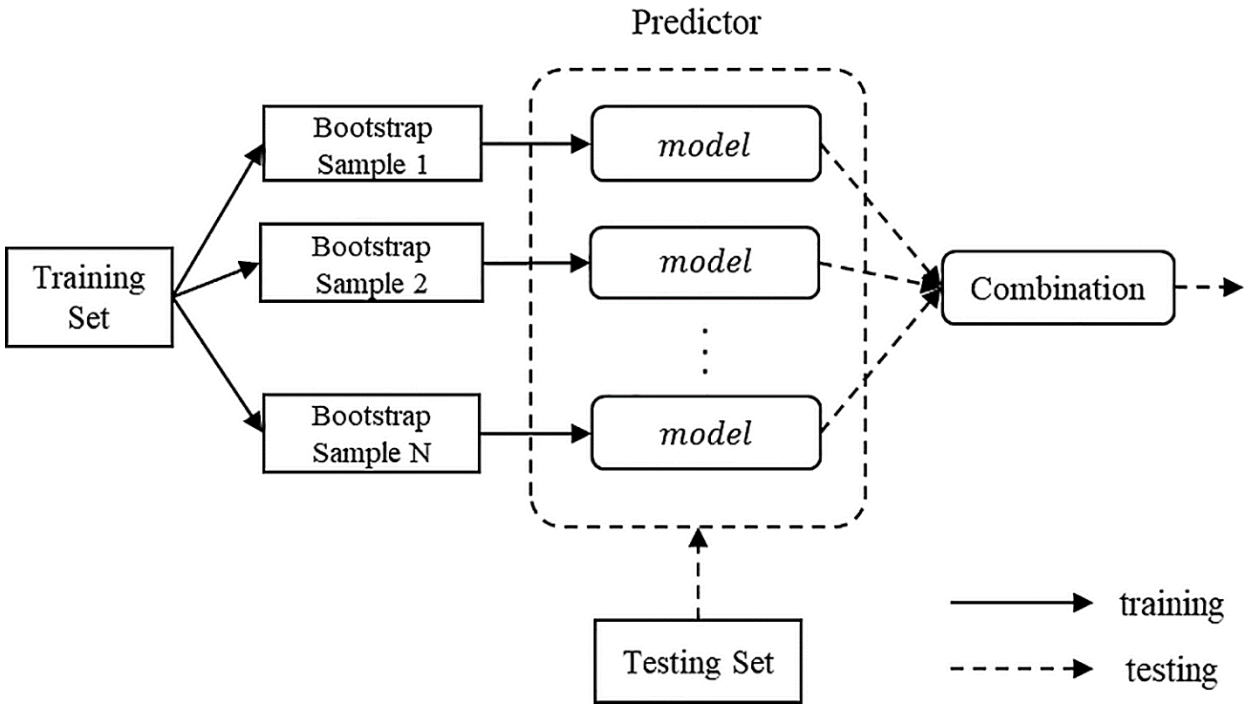


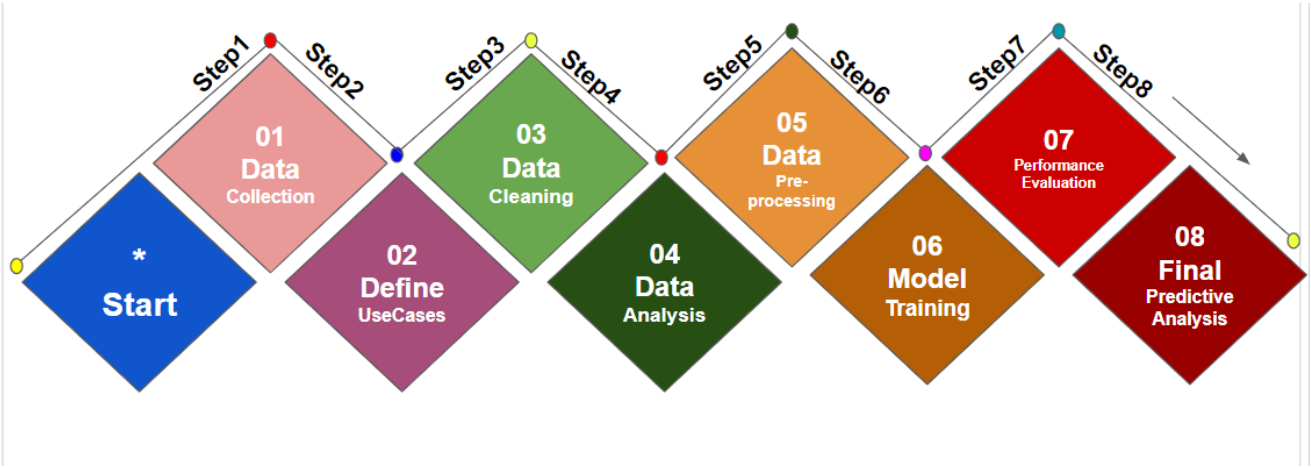


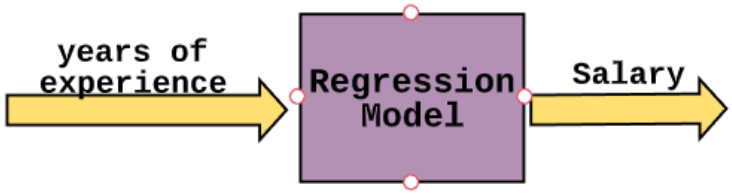
Visualize data

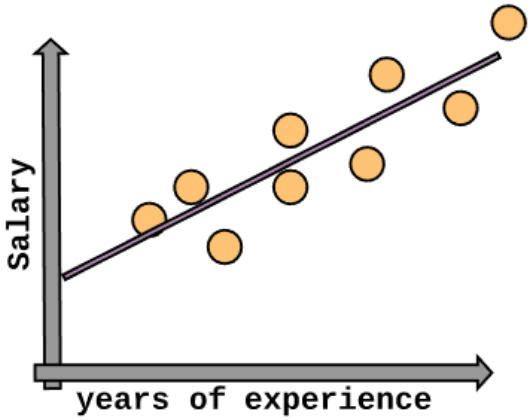
```
[6]: plt.scatter( data['YearsExperience'] ,data['Salary'] )
     plt.xlabel('Year of Exp')
     plt.ylabel('Salary')
     plt.show()
```











	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111

Define LinearRegression Model

In [77]:
```
lr = LinearRegression()
lr.fit(X_train, Y_train)
```

Out[77]:
```
LinearRegression()
```

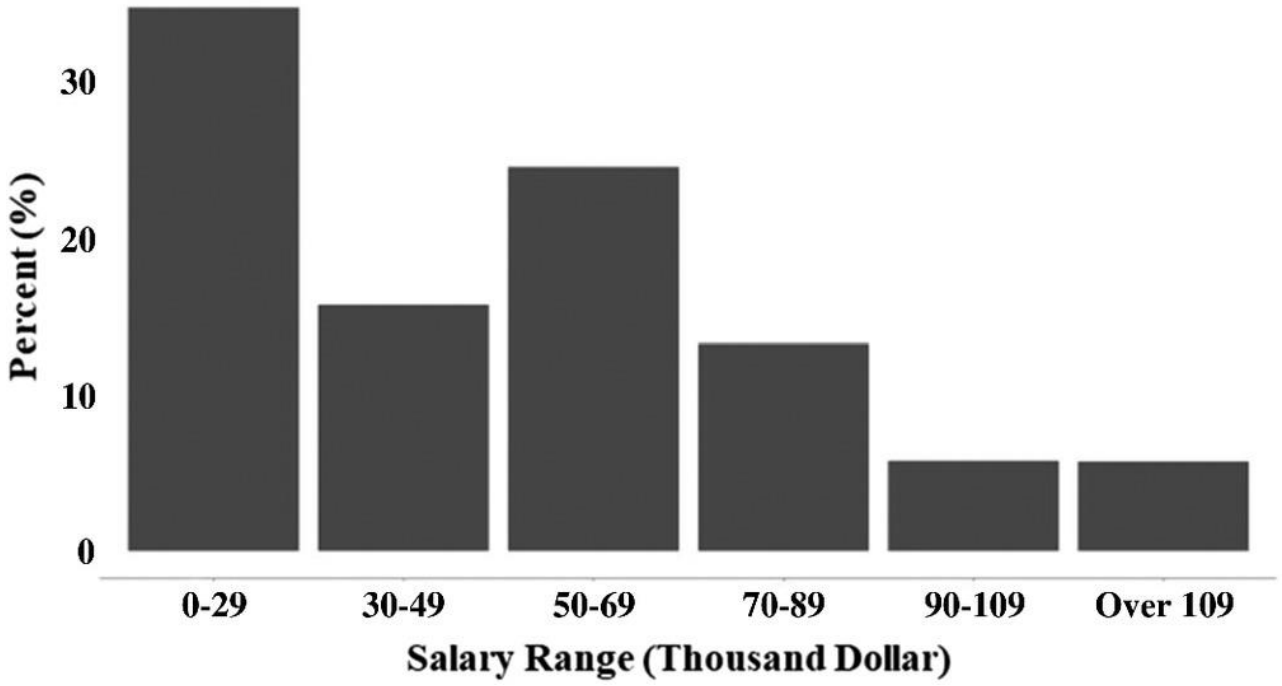
Test model

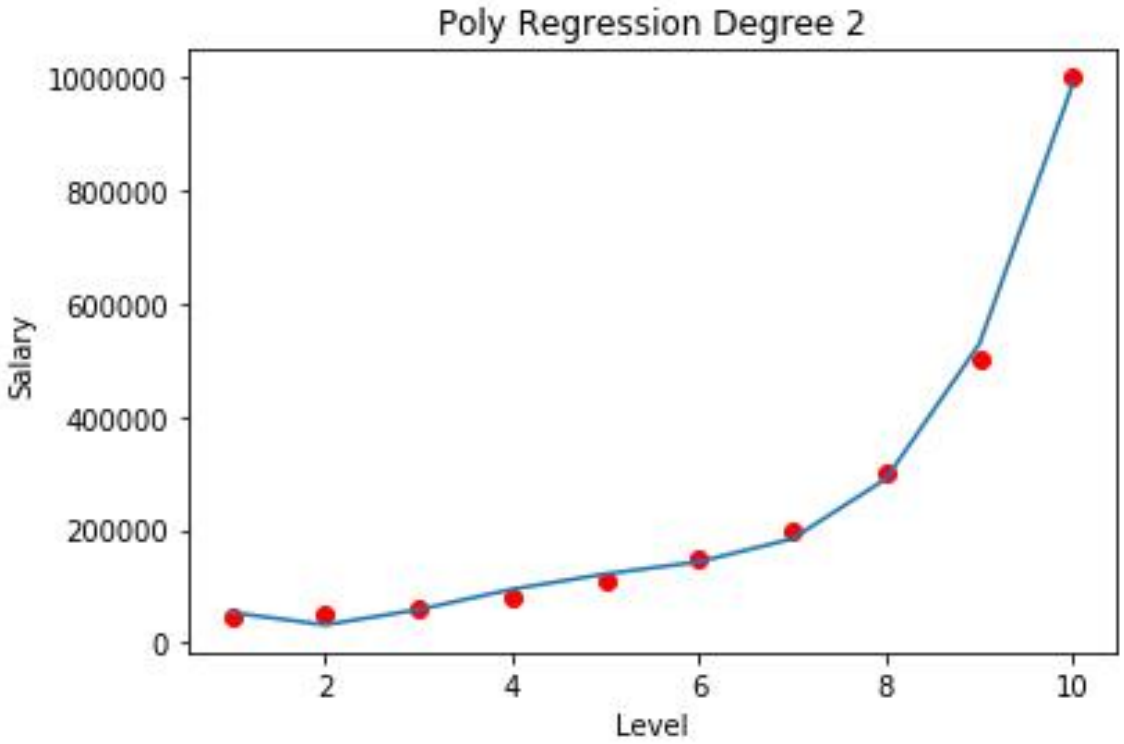
In [78]:
```
pred = lr.predict(X_test)
pred
```

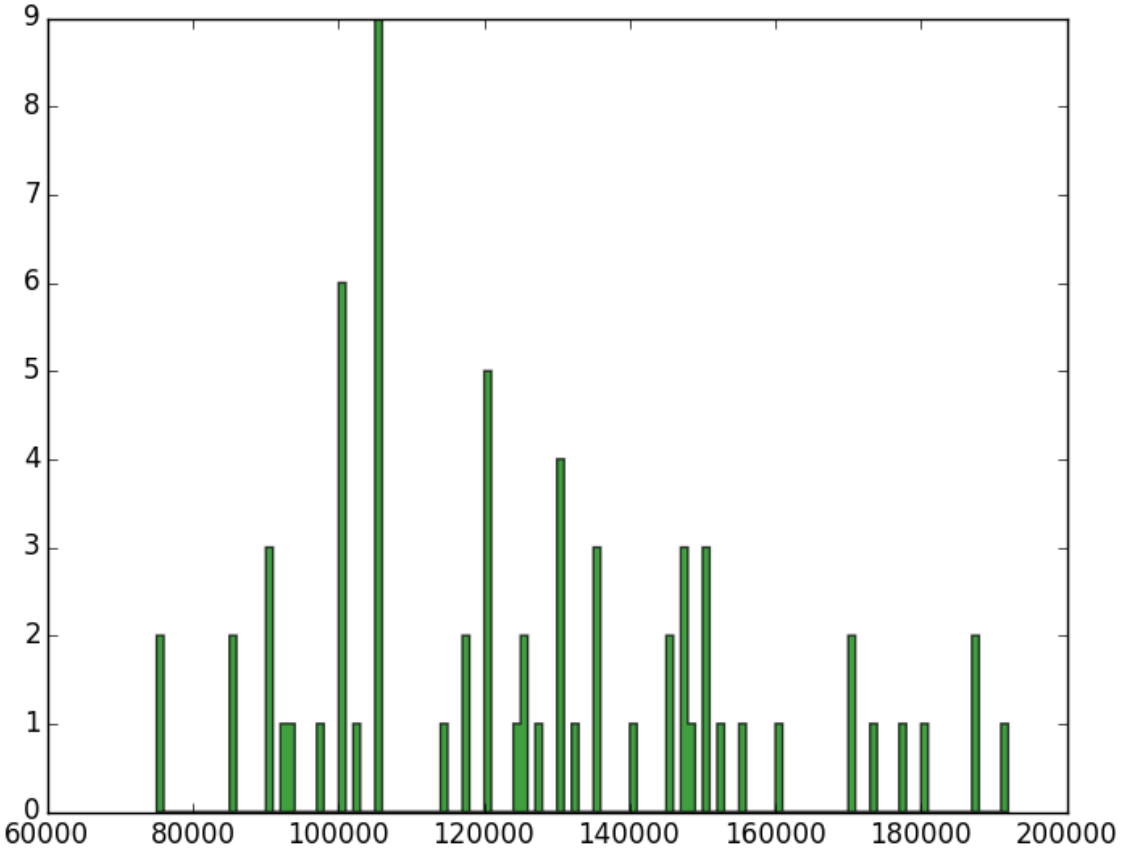
Out[78]:
```
array([ 80430.70927962,  56616.04598277,  88368.93037856, 147464.57633739,
       127178.01130675, 112183.5936754 , 107773.47084266])
```

In [68]:
```
Y_test
```

Out[68]:
```
18     81363
7      54445
20     91738
34    139465
30    127345
26    116969
25    105582
Name: Salary, dtype: int64
```




Poly Regression Degree 2
Salary
Level
1000000
800000
600000
400000
200000
0
2
4
6
8
10


9
8
7
6
5
4
3
2
1
0
60000
80000
100000
120000
140000
160000
180000
200000

RESULT

There are many different ways to build a salary prediction model, but most of them involve using machine learning techniques to analyze data about salaries and the factors that might affect them. The goal of the model is to make predictions about the salary a person is likely to earn based on the input data provided. It's important to note that the accuracy of a salary prediction model will depend on the quality of the training data and the complexity of the model. More complex models might be able to make more accurate predictions, but they may also be more difficult to understand and interpret.

CONCLUSION

salary prediction model is a tool that uses machine learning techniques to analyze data about salaries and the factors that might affect them in order to make predictions about the salary a person is likely to earn. The accuracy of the model will depend on the quality of the training data and the complexity of the model. Such a model can be useful for employers in determining salary ranges for job openings and for job seekers in understanding what salary they might expect for a given position. However, it is important to note that salary prediction models are not perfect and should not be the only factor considered when making decisions about hiring or negotiating salary.

REFERENCES

1. Predictive modeling of employee salary based on various attributes" by M. Shah, R. Shukla, and P. Shah

2. Predicting employee salaries using machine learning" by D. Li and J. Li