A PROJECT REPORT

on

"PREDICTION OF SALRY"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER & SCIENCE ENGGNERING

BY

AAYUSH RAUNIYAR	1905344
KSHITIJ BOHARA	1905280
KISHAN RAJ RAUNIYAR	1905894

UNDER THE GUIDANCE OF

AJAY ANAND

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

APRIL 2022

A PROJECT REPORT

on

"Predicting the Salary"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

Aayush Rauniyar	1905344
Kshitij Bohara	1905280
Kishan Raj Rauniyar	1905894

UNDER THE GUIDANCE OF
AJAY ANAND

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

APRIL 2022

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"PREDICTING SALARY"

submitted by

AAYUSH RAUNIYAR	1905344
KSHITIJ BOHARA	1905280
KISHAN RAJ RAUNIYAR	1905894

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 04 / 15 / 2022

(Ajay Anand)
Project Guide

Acknowledgements

We are profoundly grateful to **AJAY ANAND** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. .....

Aayush Rauniyar
Kshitij Bohara
Kishan Raj Rauniyar

ABSTRACT

Abstract: - Machine learning is a technology which allows a software program to became more accurate at pretending more accurate results without being explicitly programmed and also ML algorithms uses historic data to predicts the new outputs. Because of this ML gets a distinguish attention. Now a day's prediction engine has become so popular that they are generating accurate and affordable predictions just like a human, and being using industry to solve many of the problems. Predicting justified salary for employee is always being a challenging job for an employer. In this paper and proposing a salary prediction model with suitable algorithm using key features required to predict the salary of employee.

Now days, Major reason an employee switches the company is the salary of the employee. Employees keep switching the company to get the expected salary. And it leads to loss of the company and to overcome this loss we came with an idea what if the employee gets the desired/expected salary from the Company or Organization. In this Competitive world everyone has a higher expectation and goals. But we cannot randomly provide everyone their expected salary there should be a system which should measure the ability of the Employee for the Expected salary. We cannot decide the exact salary but we can predict it by using certain data sets. A prediction is an assumption about a future event. In this paper the main aim is predicting salary and making a suitable user-friendly graph. So that an Employee can get the desired salary on the basis of his qualification and hard work. For developing this system, we are using a Linear regression algorithm of supervised learning in machine learning

Keywords: Linear Regression.Grid SearchCV , Decision Tree and Random Search Tree.

Contents

Chapter 1

Introduction

Machine Learning have been great changes of course in Data Analysis Field . We have been implementing a lot of Machine Learning models to get and accurate results. In the current project we address one of the using Machine computation to predict system, involved in analysis the give data and refining to give required or predict the Outcomes of certain data. Namely, we are interested in predicting the salary of any person on there basis of years of experience ,qualification, and the region of there job. Possible applications for this would include prediction of any price value of any Cryptocurrency or any changes in Stock market on the basis of the past records , which will be maximized to greater extent to make good amount of profit in future.

Machine learning evolved from the study of pattern recognition and explores the notion that algorithms can learn from and make predictions on data. And, as they begin to become more ‘intelligent’, these algorithms can overcome program instructions to make highly accurate, data-driven decisions.

Predictive analytics is driven by predictive modelling. It’s more of an approach than a process. Predictive analytics and machine learning go hand-in-hand, as predictive models typically include a machine learning algorithm. These models can be trained over time to respond to new data or values, delivering the results the business needs. Predictive modelling largely overlaps with the field of machine learning.So we have used these models and compared the data from each and models after training them with data

We have used the following models :

- Random Forest Regressor
- Linear Regression
- Decision Tress Regressor

In this project we have divided this problem “Prediction of the Salary” is divided into three parts:

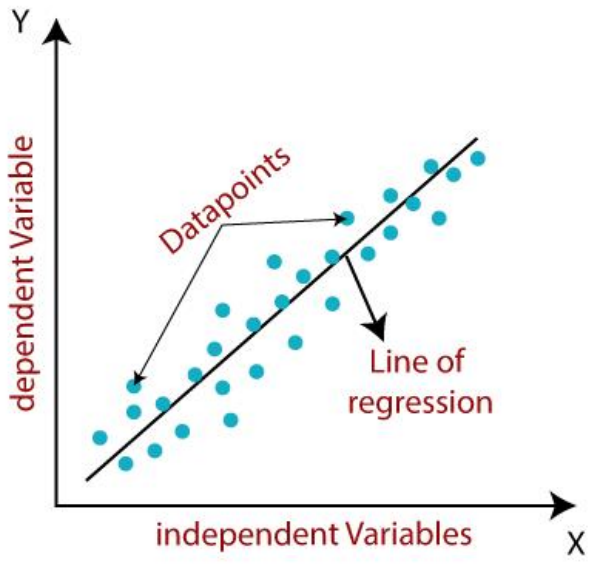
a) Cleaning the data
b) Training the data in different models
c) Exporting this data into an web application

Chapter 2

Basic Concepts/ Literature Review

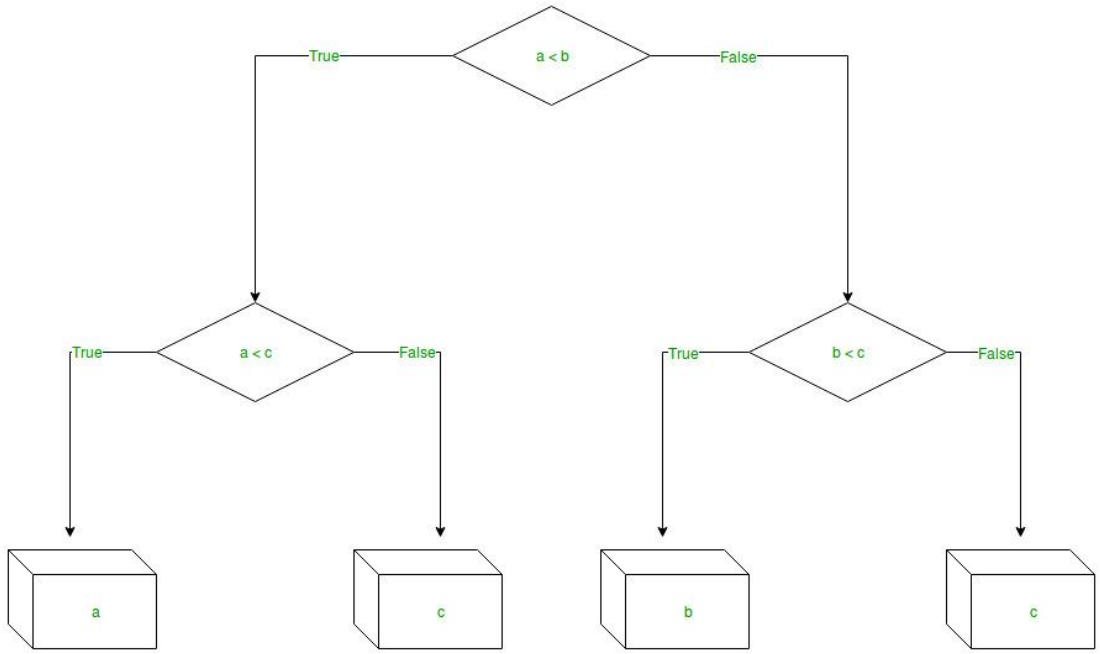
2.1 Linear Regression

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.



2.2 Decision Tress

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.



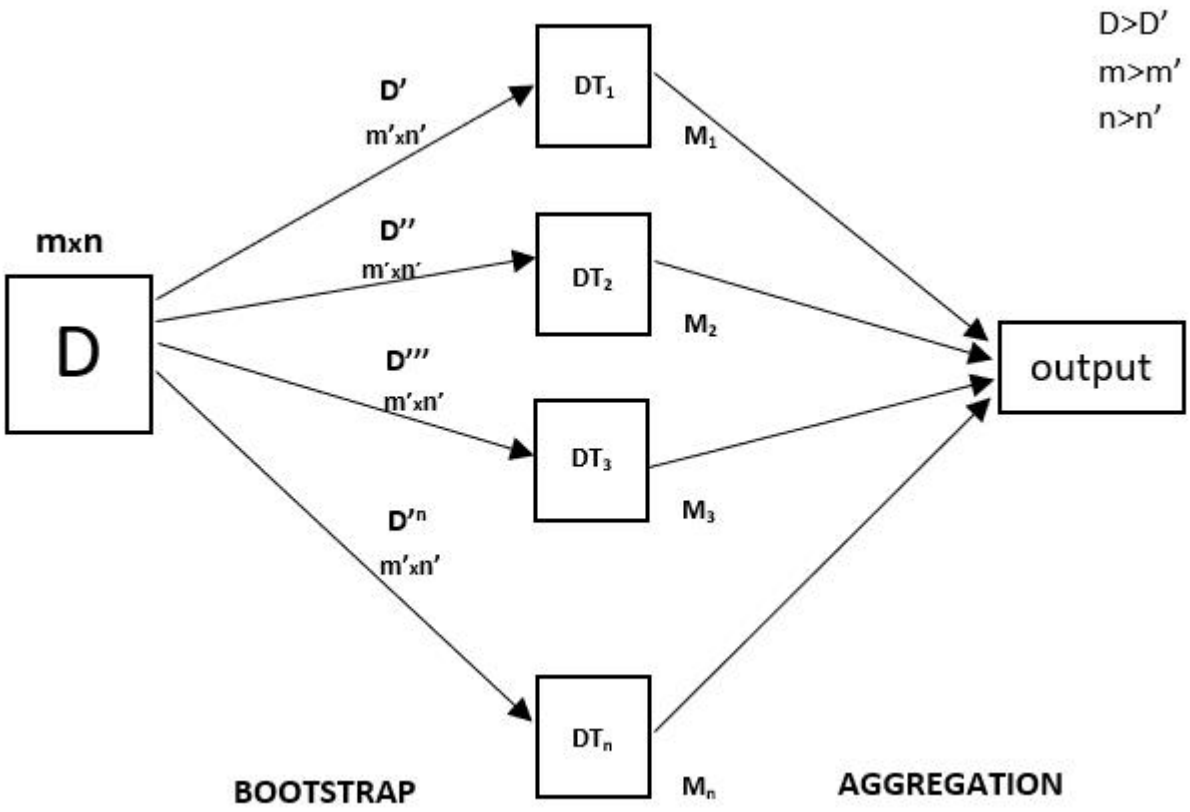
2.2.1 Decision Tree Regression Algorithm :

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values

2.3 Random Forest Regression Algorithm:

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that

particular sample data and hence the output doesn't depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation.



A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the data set forming sample datasets for every model. This part is called Bootstrap.

Chapter 3

Problem Statement / Requirement Specifications

Proposed Method

3.1 Algorithm

We have used the algorithm which uses the underlying principles of linear regression, decision tree and random forest as the project we choose is of prediction of data. For this, we have calculated the mean square error for results obtained from the machine learning model and choose the best among them.

1. Data Cleaning: For this project we have used the data collected by StackOverflow for the year 2020 from different users based on their country, age, education level, skill, salary and many more. In this part we are removing rows having null entries and removing all other columns which are not helpful in this matter.

2. Data Training on Machine Learning Model:
We try out different machine learning models and for this we use the scikit-learn library and in this case we deal with a regression problem and this means that here we don't predict a defined category like cat or dog but here we predict a number so that's why this is a regression. So first we used the linear regression model on this dataset and created a regression model.Normally,we need to split our data into training and testing or even training validation and testing data and then do the training or the fitting only on the training data but here for simplicity we do this on the whole data so now we get the predictions back. In order to evaluate how good our model

performs so for the regression plot problem we calculated mean squared error. For convinence we have set the bar for salary from $10,000 to $250,000 per annum. Since linear regression model is not effecient so we tried different model i.e decision tree regressor. Again, we calaculate root mean squared error which comes less than that of linear regression model but again it is not satisfactiory enough. So we choosed another machine learning model i.e random forest regressor in which we basically combine multiple decision trees into a forest Again, we applied same process for error calculation and this gives better result than other two models but still we were not satisfied with the result.

So, We used GridSearchCV function to loop through predefined hyperparameters and fit our estimator (model) on our training dataset. This function returns the best model for us with suitable parameter in it.

Chapter 4

Implementation

The above proposed algorithm was implemented using Python in the form of a web application having a GUI and made using the streamlit. The application is ”pip” installable and can be run through a single command from the terminal. Through the use of streamlit, the GUI can be run.

4.1 Software Requirements

• Currently the application is supported on Windows operating system.
• Python version >= 3.9 is required.
• Python modules include numpy, pandas, matplot-lib, pickle and scikit-learn

4.2 Installation

If you do not want the package to be installed globally in your system:
• Create a virtual environment so that the changes does not affect your system:
python3 -m venv <env-name>
• Activate the environment: . <env-name>/bin/activate
• Install the package using pip: pip install <path-to-folder>
i. pip install streamlit
ii. pip install scikit-learn
iii. pip install matplotlib
iv. pip install numpy
v. pip install pandas

4.3 Interface

For the interface ,we have developed web application using Streamlit and Pickel python Library

4.3.1 Streamlit

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc. With Streamlit, no callbacks are needed since widgets are treated as variables.We have used this as interface and display the results from our trained model stored in the PKL file.

4.3.1 Pickel Python

Python pickle module is used for serializing and de-serializing a Python object structure.What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.We have serialized the our train data set into file and the de-serialized it and displayed the PKL with help of streamlit.

4.4 Working:

The web application is the interface of this project.It has two different sections that are:

- ✓ **Prediction Page**
- ✓ **And Explore Page**

Initially *App.py* python file runs in which it calls other different modules and other python files.The Page initially is showing the option whether the user wants to predict the salary on the basis of Region,Years of Experience and Education or they want to explore the overall data in which data is represented in the Pie-chart or Bar Graph.After the Selection is done by the user.*App.py* calls that modules to perform the necessary task.

Predict_page.py is the Prediction modules of our project.In this module we have made the function *load_model()*to load the PKL file created after data with help different regression model and the store that trained dataset into "data "variable.After that we have *show_predict_page( )* function to predict the salary using trained model.In this function we have list of different countries and education degree.From that list user have to select particular country and education level which will be stored in the respective variable *country()* and *education()* and for the experience we have made Slide Bar to keep unique and it will be stored in the variable *expericence()*.After Pressing the "Calculate Salary" Button.All this variable data is transferred into *numpy Array* X and regressor model is used to predict the salary using X variable and shown the output using the streamlit Sub Header Tag.

Explore_page.py is the exploration module of a overall data ans statistic of the refined data along with Bar Graph for Mean Salary Based on Experience and Mean Salary Based on Country and Pie chart for the number of data from all other different countries.For that we have 1st we have to clean the data and we have two function modules *clean_experience(x)* and *clean_education(x).* In the *clean_experience(x)* we have separated the value of experience value 'More than 50 years' and 'Less than 1 year'. And another function is to categories there education qualification.Now the *load_data()* is the main function in we load the csv file and from that we can extract the 'ConvertedComp' and 'Country'.

And from that CSV file we have extracted the data ranging from 250000 to 10000 and renaming the 'ConvertedComp' as Salary and saving this whole data in *load_data()*. And in the function *show_explore_data ()* we have plotted the Bar Graph for Mean Salary Based on Experience and Mean Salary Based on Country and Pie chart for the number of data from all other different countries.

Chapter 5

Screenshots of Project

5.1 Snippets Of The Code along with testing models

```
[22]
from sklearn.linear_model import LinearRegression
linear_reg = LinearRegression()
linear_reg.fit(X, y.values)

LinearRegression()

[23]
y_pred = linear_reg.predict(X)

[24]
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
error = np.sqrt(mean_squared_error(y, y_pred))

[25]
error

39274.75368318509
```

Fig : Linear Regression Model

```
[26]
from sklearn.tree import DecisionTreeRegressor
dec_tree_reg = DecisionTreeRegressor(random_state=0)
dec_tree_reg.fit(X, y.values)

DecisionTreeRegressor(random_state=0)

[27]
y_pred = dec_tree_reg.predict(X)

[28]
error = np.sqrt(mean_squared_error(y, y_pred))
print("${:,.02f}".format(error))

$29,414.94
```

Fig : Decision Tree Regression Model

```
[29]
    from sklearn.ensemble import RandomForestRegressor
    random_forest_reg = RandomForestRegressor(random_state=0)
    random_forest_reg.fit(X, y.values)

...  RandomForestRegressor(random_state=0)

[30]
    y_pred = random_forest_reg.predict(X)

[31]
    error = np.sqrt(mean_squared_error(y, y_pred))
    print("${:,.02f}".format(error))

...  $29,487.31
```

Fig : Random Forest Regression Model

```
from sklearn.model_selection import GridSearchCV

max_depth = [None, 2,4,6,8,10,12]
parameters = {"max_depth": max_depth}

regressor = DecisionTreeRegressor(random_state=0)
gs = GridSearchCV(regressor, parameters, scoring='neg_mean_squared_error')
gs.fit(X, y.values)
```

```
GridSearchCV(estimator=DecisionTreeRegressor(random_state=0),
             param_grid={'max_depth': [None, 2, 4, 6, 8, 10, 12]},
             scoring='neg_mean_squared_error')
```

```
regressor = gs.best_estimator_

regressor.fit(X, y.values)
y_pred = regressor.predict(X)
error = np.sqrt(mean_squared_error(y, y_pred))
print("${:,.02f}".format(error))
```

```
$30,428.51
```

Fig : Grid Search CV Model

```
X
```

	Country	EdLevel	YearsCodePro
7	13	0	13.0
9	12	2	4.0
10	12	0	2.0
11	10	1	7.0
12	7	1	20.0
...	...	...	...
64113	13	1	15.0
64116	13	0	6.0
64122	13	1	4.0
64127	13	3	12.0
64129	13	2	4.0

18491 rows × 3 columns

```
# country, edlevel, yearscode
X = np.array([["United States", 'Master's degree', 15 ]])
X
```

```
array([['United States', 'Master's degree', '15']], dtype='<U15')
```

```
X[:, 0] = le_country.transform(X[:,0])
X[:, 1] = le_education.transform(X[:,1])
X = X.astype(float)
X
```

```
array([[13.,  2., 15.]])
```

```
y_pred = regressor.predict(X)
y_pred
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
DecisionTreeRegressor was fitted with feature names
  warnings.warn(

array([139427.26315789])
```

Fig : Grid Search CV showing which model best fit.

```
import pickle
data = {"model": regressor, "le_country": le_country, "le_education": le_education}
with open('saved_steps.pkl', 'wb') as file:
    pickle.dump(data, file)
```

```
with open('saved_steps.pkl', 'rb') as file:
    data = pickle.load(file)

regressor_loaded = data["model"]
le_country = data["le_country"]
le_education = data["le_education"]
```

```
y_pred = regressor_loaded.predict(X)
y_pred
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
DecisionTreeRegressor was fitted with feature names
  warnings.warn(

array([139427.26315789])
```

Fig:Pickel UI code

app.py > ...

```
import streamlit as st
from predict_page import show_predict_page
from explore_page import show_explore_page


page = st.sidebar.selectbox("Explore Or Predict", ("Predict", "Explore"))

if page == "Predict":
    show_predict_page()
else:
    show_explore_page()

```

Fig : Main python

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

def shorten_categories(categories, cutoff):
    categorical_map = {}
    for i in range(len(categories)):
        if categories.values[i] >= cutoff:
            categorical_map[categories.index[i]] = categories.index[i]
        else:
            categorical_map[categories.index[i]] = 'Other'
    return categorical_map
def clean_experience(x):
    if x ==  'More than 50 years':
        return 50
    if x == 'Less than 1 year':
        return 0.5
   return float(x)
def clean_education(x):
    if 'Bachelor's degree' in x:
        return 'Bachelor's degree'
    if 'Master's degree' in x:
        return 'Master's degree'
    if 'Professional degree' in x or 'Other doctoral' in x:
        return 'Post grad'
    return 'Less than a Bachelors'
@st.cache
def load_data():
    df = pd.read_csv("survey_results_public_2020.csv")
    df = df[["Country", "EdLevel", "YearsCodePro", "Employment",
"ConvertedComp"]]
    df = df[df["ConvertedComp"].notnull()]
    df = df.dropna()
    df = df[df["Employment"] == "Employed full-time"]
    df = df.drop("Employment", axis=1)
    country_map = shorten_categories(df.Country.value_counts(), 400)
    df["Country"] = df["Country"].map(country_map)
    df = df[df["ConvertedComp"] <= 250000]
    df = df[df["ConvertedComp"] >= 10000]
    df = df[df["Country"] != "Other"]
    df["YearsCodePro"] = df["YearsCodePro"].apply(clean_experience)
    df["EdLevel"] = df["EdLevel"].apply(clean_education)
    df = df.rename({"ConvertedComp": "Salary"}, axis=1)
    return df
df = load_data()
def show_explore_page():
    st.title("Explore Software Engineer Salaries")
    st.write(
        """
    ### Stack Overflow Developer Survey 2020
    """
    )
    data = df["Country"].value_counts()
    fig1, ax1 = plt.subplots()
```

```
    ax1.pie(data, labels=data.index, autopct="%1.1f%%", shadow=True,
startangle=90)
    ax1.axis("equal")      st.write("""#### Number of Data from different
countries""")
    st.pyplot(fig1)

    st.write(
        """
    #### Mean Salary Based On Country
    """
    )
    data =
df.groupby(["Country"])["Salary"].mean().sort_values(ascending=True)
    st.bar_chart(data)
    st.write(
        """
    #### Mean Salary Based On Experience
    """
    )
    data =
df.groupby(["YearsCodePro"])["Salary"].mean().sort_values(ascending=True)
    st.line_chart(data)
```

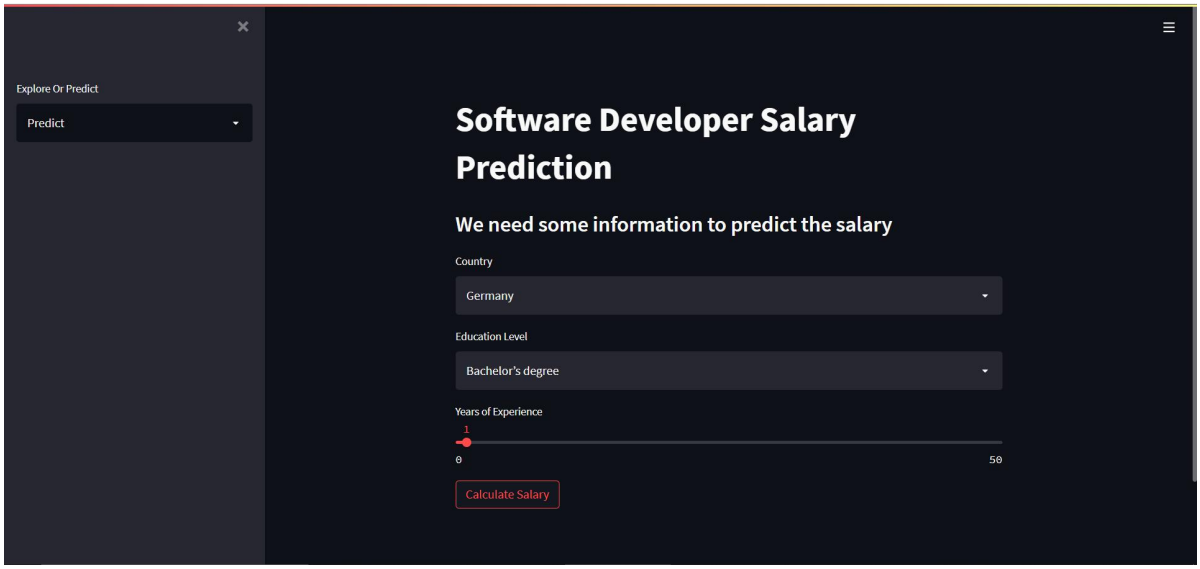
Fig : Code of Explore App

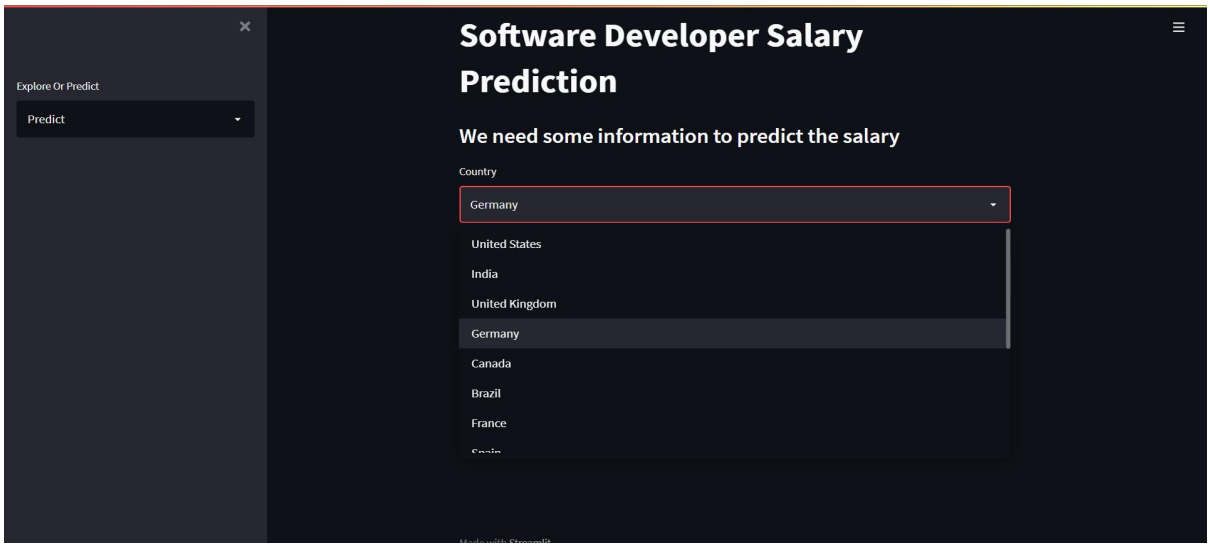
```
import streamlit as st
import pickle
import numpy as np


def load_model():
    with open('saved_steps.pkl', 'rb') as file:
        data = pickle.load(file)
    return data
data = load_model()
regressor = data["model"]
le_country = data["le_country"]
le_education = data["le_education"]
def show_predict_page():
    st.title("Software Developer Salary Prediction")
    st.write("""### We need some information to predict the salary""")
    countries = (
        "United States",
        "India",
        "United Kingdom",
        "Germany",
        "Canada",
        "Brazil",
        "France",
        "Spain",
        "Australia",
        "Netherlands",
        "Poland",
        "Italy",
        "Russian Federation",
        "Sweden",
    )
    education = (
        "Less than a Bachelors",
```

```
        "Bachelor’s degree",
        "Master’s degree",
        "Post grad",
    )
    country = st.selectbox("Country", countries)
    education = st.selectbox("Education Level", education)
    expericence = st.slider("Years of Experience", 0, 50, 3)
    ok = st.button("Calculate Salary")
    if ok:
        X = np.array([[country, education, expericence ]])
        X[:, 0] = le_country.transform(X[:,0])
        X[:, 1] = le_education.transform(X[:,1])
        X = X.astype(float)
        salary = regressor.predict(X)
        st.subheader(f"The estimated salary is ${salary[0]:.2f}")
```

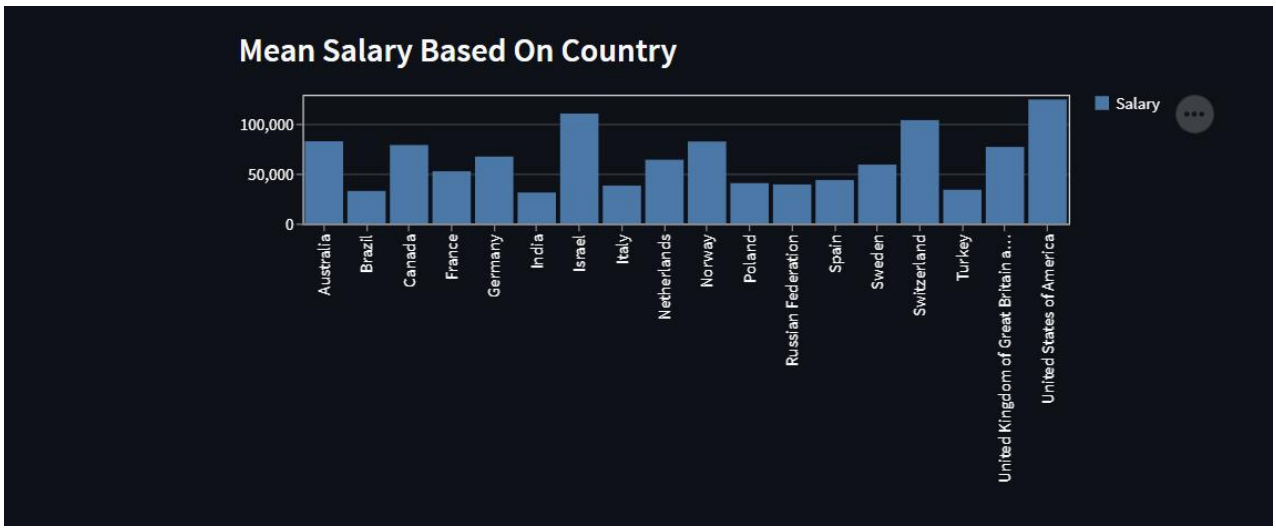
Fig : Code of Predict App

5.2 Prediction Results






Explore Or Predict
Explore
Mean Salary Based On Experience
Salary
160,000
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0
0
5
10
15
20
25
30
35
40
45
50


Mean Salary Based On Country
Salary
100,000
50,000
0
Australia
Brazil
Canada
France
Germany
India
Israel
Italy
Netherlands
Norway
Poland
Russian Federation
Spain
Sweden
Switzerland
Turkey
United Kingdom of Great Britain a...
United States of America

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

The approach presented here is the Usage for the Machine Learning Prediction using the data and predict the future valuation of any object.Predicting the Values is done using **Decision Tress Regressor** algorithm where as **Grid Search CV** is used to detect which one algorithm is best giving less deviation from actual data.Using this system many good predicting systems can be developed and required object can be maximize the profit easily. In the coming days these algorithms can be used to predict the price of any object like crypto Currency or stock market.

6.2 Future Scope

Future work is to work on the same domain but to predict a particular value of any object on the basis of data provided by user .

2. **Decision Tress Regressor** Algorithm is predicting but actually it is deviating more than 30000 from the Actual Mean . We would try to train classifiers so that it is subtle to all sort of predicting.

References

[1] Scikit Learn
[2] Pandas
[3] Numpy
[4] Linear Regression in Python
[5] Decision Tree
[6] Random forest
[7] Matplotlib
[8] Streamlit