A Course Based Project Report on

SALARY PREDICTION OF AN EMPLOYEE

Submitted to the

Department of CSE-(CyS, DS) and AI&DS

in partial fulfilment of the requirements for the completion of course

PYTHON PROGRAMMING LABORATORY (22ES2DS101)

BACHELOR OF TECHNOLOGY

IN

CSE- Data Science

Submitted by

CH.SEERSHIKA	23071A6715
M.SREEVIBHA	23071A6738
P.SRIVALLI	23071A6744
P.SRI VARSHA	23071A6751

Under the guidance of

Mr. G. Sathar

Assistant Professor

Department of CSE-(CyS, DS) and AI&DS

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

NOVEMBER-2024

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as "College with Potential for Excellence" by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

Department of CSE-(CyS, DS) and AI&DS

CERTIFICATE

This is to certify that the project report entitled "**Salary Prediction of an Employee**" is a bonafide work done under our supervision and is being submitted by **Miss.P.Seershika (23071A6715) , Miss. M. Sreevibha (23071A6738), Miss. P. Srivalli (23071A6744), Miss. P. Sri Varsha (23071A6751)** in partial fulfilment for the award of the degreeof **Bachelor of Technology** in **CSE-Data Science**, of the VNRVJIET, Hyderabad during the academic year 2024-2025.

Mr. G. Sathar
Assistant Professor
Dept of CSE-(CyS, DS) and AI&DS

Dr.T.SUNIL KUMAR
Professor & HOD
Dept of CSE-(CyS, DS)and AI&DS

Course based Projects Reviewer

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade,
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

Department of CSE-(CyS, DS) and AI&DS

DECLARATION

We declare that the course based project work entitled "**SALARY PREDICTION OF AN EMPLOYEE**" submitted in the Department of **CSE-(CyS, DS) and AI&DS**, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in CSE-Data Science** is a bonafide record of our own work carried out under the supervision of **Mr. G. Sathar, Assistant Professor, Department of CSE-(CyS, DS) and AI&DS , VNRVJIET.** Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

Ch.Seershika	**M.Sreevibha**	**P.Srivalli**	**P.Sri Varsha**
(23071A6715)	(23071A6738)	(23071A6744)	(23071A6751)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, **Sri. D. Suresh Babu,** VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, **Dr. C.D Naidu,** for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor **Dr. T. Sunil Kumar**, Professor and Head, Department of CSE-(CyS, DS) and AI&DS , VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our belove Guide, **Dr. G. Sathar**, Assistant Professor in CSE-(CyS, DS) and AI&DS, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

CH. SEERSHIKA	23071A6715
M. SREEVIBHA	23071A6738
P. SRIVALLI	23071A6744
P. SRI VARSHA	23071A6751

TABLE OF CONTENTS
(INDEX IN TABLE FORMAT)

ABSTRACT

The **Salary Prediction of an Employee** System is a machine learning-powered application designed to estimate an employee's salary based on key factors like experience, education level, job role, location, and industry type. This innovative system addresses the need for an accurate, efficient, and user-friendly platform that facilitates salary estimation, benefiting both employees and HR professionals in decision-making processes.

The system operates by accepting relevant inputs from users through a simple, interactive interface. These inputs are processed using a robust regression-based machine learning model trained on a comprehensive dataset. The model leverages patterns in the data to provide reliable salary predictions. The backend is implemented using Python with Flask, ensuring seamless communication between the user interface and the machine learning model. A relational database is integrated to securely store user data and predictions, enabling trend analysis and enhancing the overall utility of the system.

For HR professionals, the system serves as a valuable tool for benchmarking salaries and analyzing market trends. Employees can use the platform to understand their earning potential based on their qualifications and job-specific attributes. Administrators can update the underlying machine learning model with new datasets to maintain its accuracy and relevance over time.

Key features of the Salary Prediction System include a user-friendly input interface, dynamic salary prediction, secure data storage, and model customization capabilities. Additionally, the system is designed to be scalable, supporting a wide range of industries and job roles, making it versatile and adaptable to diverse organizational needs.

This system streamlines the process of salary estimation, reducing manual effort and providing reliable insights. It promotes transparency and informed decision-making, thereby enhancing the efficiency of compensation management processes. Through its integration of machine learning and a robust backend, the Salary Prediction System is a valuable asset for modern workplaces.

CHAPTER-1

INTRODUCTION

The **Salary Prediction of an Employee** project is an advanced analytical solution aimed at revolutionizing how organizations forecast employee compensation. By leveraging machine learning techniques, particularly Linear Regression, this project provides a robust, scalable, and efficient method to predict salaries based on measurable factors like years of experience, education level, and job role. In a rapidly evolving workforce landscape, this project addresses the need for data-driven decisions in compensation planning, fostering fairness and transparency.

Organizations across industries rely heavily on salary benchmarking and prediction to maintain competitive compensation strategies. These practices are essential to attract and retain top talent, ensure equity, and optimize budget allocations. However, traditional methods of salary estimation often involve manual analysis, which is time-consuming and prone to biases. Such approaches also lack the scalability required to handle large datasets or provide real-time insights, limiting their effectiveness in fast-paced, data-centric environments.

The Salary Prediction System overcomes these challenges by utilizing historical data and predictive modeling to deliver accurate salary forecasts. The project incorporates preprocessed datasets, encoded categorical variables, and normalized features to build a Linear Regression model that learns from existing trends and generalizes predictions for unseen data. This approach not only ensures precision but also provides actionable insights for HR professionals and decision-makers.

In conclusion, the Salary Prediction System signifies a paradigm shift in how salaries are estimated and analyzed. By automating the prediction process, it empowers organizations to make informed, equitable, and data-backed decisions. This solution bridges the gap between raw data and actionable insights, setting a new benchmark for efficiency, accuracy, and scalability in workforce analytics.

CHAPTER-2

METHOD

Development Process of the Salary Prediction System

The Salary Prediction System is a Python-based machine learning application designed to estimate employee salaries based on features like years of experience, education level, and job title. By leveraging tools such as Pandas, NumPy, Scikit-learn, and Streamlit, the project provides a data-driven solution to assist organizations in designing fair compensation strategies. Salary prediction models like this are essential for modern HR practices, as they streamline decision-making, reduce biases, and ensure consistency across various employee categories. This report outlines the structured approach followed in the development of the Salary Prediction System, covering data preprocessing, model implementation, interface design, and testing. The system aims to deliver a user-friendly and accurate platform for predicting employee salaries based on historical data storage is incorporated, and an admin module is created to manage quizzes, questions, and user data effectively.

Once the application is built, rigorous **testing** is conducted to ensure it meets the desired quality standards. Unit testing is performed on individual modules like login, quiz selection, and result calculation, while integration testing ensures seamless communication between the modules. The system is tested using sample data to verify its accuracy, usability, and reliability.

The final step is **deployment**, where the application is made available for actual use. It is either installed on target machines or deployed as a web-based solution accessible through a browser. Users are provided with instructions to access and operate the system efficiently.

To ensure the system remains functional and relevant, **maintenance and future enhancements** are prioritized. Regular updates to the database and application are performed to keep the system scalable and secure. User feedback is collected to

identify areas for improvement, with potential future features including timer-based quizzes, leaderboards, and mobile app compatibility.

This structured methodology ensures that the **Salary prediction of an Employee system** is developed efficiently and delivers a user-friendly solution for conducting quizzes, addressing the needs of students, educators, and administrators.

Data Preparation and Preprocessing

The dataset used for this project was sourced from [e.g., Kaggle], containing records of employees with features such as:

Years of Experience – Numeric data representing the employee's work history.

Education Level – Categorical data, including levels like Bachelor's, Master's, and Ph.D.

Job Title – Roles held by employees in various industries.

Salary – The target variable, representing the annual income of employees.

To ensure the data was clean and suitable for modeling, preprocessing steps were applied:

Handling Missing Values: Missing entries in critical columns were imputed using mean or mode values, depending on the data type.

Encoding Categorical Features: Categorical data such as job titles and education levels were encoded into numerical values using one-hot encoding. This step transformed the data into a format compatible with machine learning algorithms.

Normalization: Continuous variables, such as years of experience, were scaled using Min-Max normalization to ensure consistency in feature ranges. This improved the model's learning efficiency and prediction accuracy.

Model Development

The Salary Prediction System employs Linear Regression, a fundamental machine learning algorithm that models the relationship between independent variables and a continuous dependent variable. This algorithm was chosen for its simplicity, interpretability, and effectiveness in handling structured data.

Training and Testing

The dataset was split into two parts:

Training Set (80%): Used to train the model and adjust the regression coefficients.

Testing Set (20%): Used to evaluate the model's performance on unseen data.

The train_test_split function from Scikit-learn was used for this purpose, ensuring a random yet reproducible split.

Implementation

Using Scikit-learn's LinearRegression class, the model was trained on the preprocessed data. It learned the relationships between features like years of experience, job title, previous salary.

DATASET

	Respondent	MainBranch	Hobbyist	Age	Age1stCode	CompFreq	CompTotal	ConvertedComp	Country	CurrencyDesc	...	SurveyEase	SurveyLength	Trans
0	1	I am a developer by profession	Yes	NaN	13	Monthly	NaN	NaN	Germany	European Euro	...	Neither easy nor difficult	Appropriate in length	No
1	2	I am a developer by profession	No	NaN	19	NaN	NaN	NaN	United Kingdom	Pound sterling	...	NaN	NaN	NaN
2	3	I code primarily as a hobby	Yes	NaN	15	NaN	NaN	NaN	Russian Federation	NaN	...	Neither easy nor difficult	Appropriate in length	NaN
3	4	I am a developer by profession	Yes	25.0	18	NaN	NaN	NaN	Albania	Albanian lek	...	NaN	NaN	No

```
df = df[["Country", "EdLevel", "YearsCodePro", "Employment", "ConvertedComp"]]
df = df.rename({"ConvertedComp": "Salary"}, axis=1)
df.head()
```

	Country	EdLevel	YearsCodePro	Employment	Salary
0	Germany	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	27	Independent contractor, freelancer, or self-em...	NaN
1	United Kingdom	Bachelor's degree (B.A., B.S., B.Eng., etc.)	4	Employed full-time	NaN
2	Russian Federation	NaN	NaN	NaN	NaN
3	Albania	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	4	NaN	NaN
4	United States	Bachelor's degree (B.A., B.S., B.Eng., etc.)	8	Employed full-time	NaN

```
df = df[df["Salary"].notnull()]
df.head()
```

	Country	EdLevel	YearsCodePro	Employment	Salary
7	United States	Bachelor's degree (B.A., B.S., B.Eng., etc.)	13	Employed full-time	116000.0
9	United Kingdom	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	4	Employed full-time	32315.0
10	United Kingdom	Bachelor's degree (B.A., B.S., B.Eng., etc.)	2	Employed full-time	40070.0
11	Spain	Some college/university study without earning ...	7	Employed full-time	14268.0
12	Netherlands	Secondary school (e.g. American high school, G...	20	Employed full-time	38916.0

CHAPTER-3

CODE

PYTHON:

```
fig, ax = plt.subplots(1,1, figsize=(12, 7))
df.boxplot('Salary', 'Country', ax=ax)
plt.suptitle('Salary (US$) v Country')
plt.title('')
plt.ylabel('Salary')
plt.xticks(rotation=90)
plt.show()
```

```
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\SRI VARSHA\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:49
  warnings.warn(
```

```
array([139427.26315789])
```

```
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"]
```

```
df["YearsCodePro"].unique()
```

```
array(['13', '4', '2', '7', '20', '1', '3', '10', '12', '29', '6', '28',
       '8', '23', '15', '25', '9', '11', 'Less than 1 year', '5', '21',
       '16', '18', '14', '32', '19', '22', '38', '30', '26', '27', '17',
       '24', '34', '35', '33', '36', '40', '39', 'More than 50 years',
       '31', '37', '41', '45', '42', '44', '43', '50', '49'], dtype=object)
```

```
def clean_experience(x):
    if x ==  'More than 50 years':
        return 50
    if x == 'Less than 1 year':
        return 0.5
    return float(x)

df['YearsCodePro'] = df['YearsCodePro'].apply(clean_experience)
```

```
df["EdLevel"].unique()
```

```
array(['Bachelor's degree (B.A., B.S., B.Eng., etc.)',
       'Master's degree (M.A., M.S., M.Eng., MBA, etc.)',
       'Some college/university study without earning a degree',
       'Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)',
       'Associate degree (A.A., A.S., etc.)',
       'Professional degree (JD, MD, etc.)',
       'Other doctoral degree (Ph.D., Ed.D., etc.)',
       'I never completed any formal education',
       'Primary/elementary school'], dtype=object)
```

```
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'

df['EdLevel'] = df['EdLevel'].apply(clean_education)
```

```
df["EdLevel"].unique()
```

```
array(['Bachelor’s degree', 'Master’s degree', 'Less than a Bachelors',
       'Post grad'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
le_education = LabelEncoder()
df['EdLevel'] = le_education.fit_transform(df['EdLevel'])
df["EdLevel"].unique()
#le.classes_
```

```
array([0, 2, 1, 3])
```

```
le_country = LabelEncoder()
df['Country'] = le_country.fit_transform(df['Country'])
df["Country"].unique()
```

```
array([13, 12, 10,  7,  4,  2,  6,  1,  3,  5, 11,  8,  0,  9])
```

```
X = df.drop("Salary", axis=1)
y = df["Salary"]
```

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

```
LinearRegression()
```

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

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

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

```
DecisionTreeRegressor(random_state=0)
```

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

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

```
$29,414.94
```

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

```
RandomForestRegressor(random_state=0)
```

prediction.py:

```
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 salar

    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)
```

```
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}")
```

explore.py:

```
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_resource
def load_data():
    df = pd.read_csv("survey_results_public.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"]
```

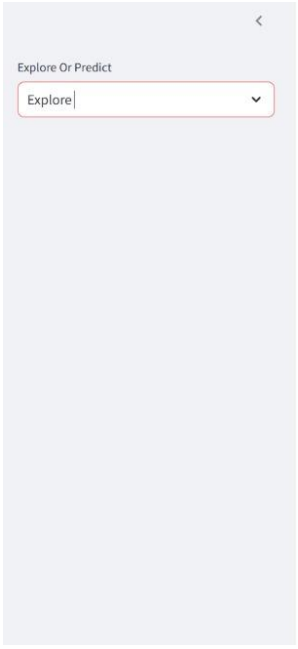
```
    df["YearsCodePro"] = df["YearsCodePro"].apply(clean_experience)
    df["EdLevel"] = df["EdLevel"].apply(clean_education)
    df = df.rename({"ConvertedComp": "Salary"}, axis=1)
    return df

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")  # Equal aspect ratio ensures that pie is drawn as a circle.

    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)
```

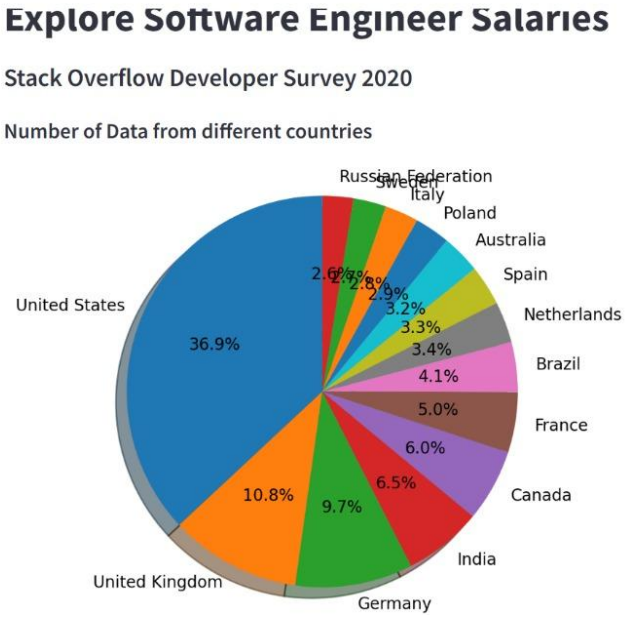
```
import streamlit as st
from predict_page import show_predict_page
from explore_page import show_explore_page


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

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

CHAPTER-4

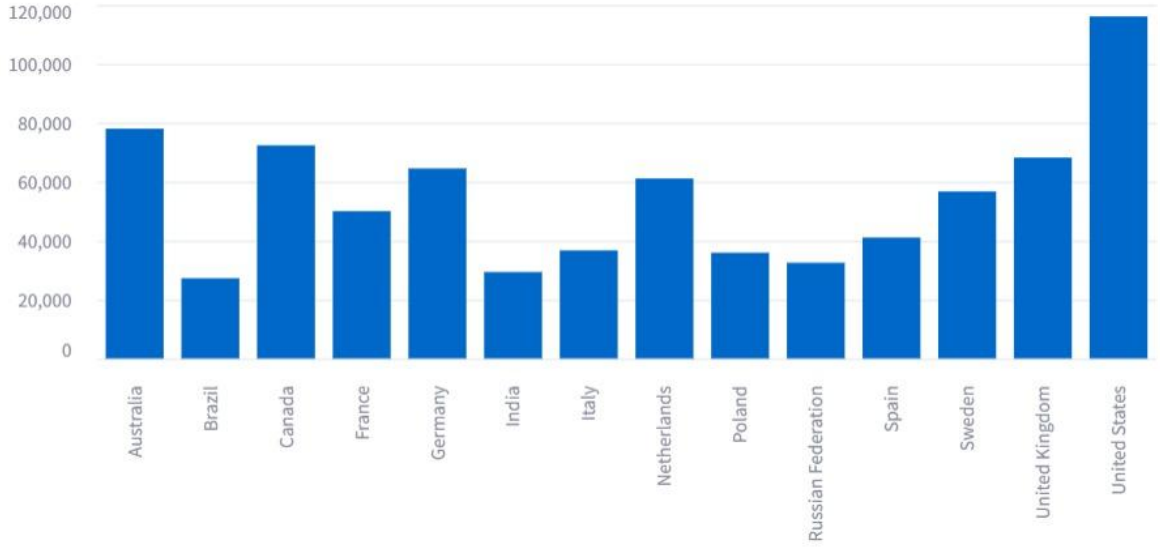
TEST CASES/ OUTPUT

Test case 1:





Mean Salary Based On Country



Mean Salary Based On Experience

Test case 2:

Test case 3:

CHAPTER-5

RESULTS

The Salary Prediction System is a robust and user-friendly platform designed to simplify the process of estimating employee salaries based on key attributes such as years of experience, education level, and job role. It leverages the power of Python and machine learning algorithms to provide accurate and data-driven salary predictions. The system allows users to input relevant employee details through an intuitive interface and instantly view the predicted salary. By automating the prediction process, the system eliminates manual estimation errors and ensures consistency in salary assessments.

The backend of the system is powered by a preprocessed dataset and a Linear Regression model trained to identify patterns and relationships in the data. Tools such as Pandas and NumPy ensure efficient data handling, while Scikit-learn provides the framework for model implementation and evaluation. A user-friendly web interface built using Streamlit allows seamless interaction, enabling users to input data and retrieve predictions effortlessly.

The system also incorporates robust data preprocessing techniques, including handling missing values, encoding categorical variables, and scaling numerical data, ensuring high accuracy and reliability in predictions. It is designed to scale, making it suitable for various applications ranging from small businesses to large-scale enterprise solutions.

With a focus on error handling, real-time predictions, and a streamlined user experience, the Salary Prediction System is well-suited for HR analytics, workforce planning, and other domains requiring accurate salary insights. The platform demonstrates scalability and adaptability, ensuring it can handle increasing volumes of data and diverse use cases without compromising performance.

CHAPTER – 6

SUMMARY

The **Salary Prediction System** is an innovative and practical solution designed to estimate employee salaries using machine learning techniques. Developed in Python, the system leverages a Linear Regression model to predict salaries based on key attributes such as years of experience, education level, and job role. This data-driven approach enhances accuracy, reduces biases, and simplifies the traditionally manual process of salary estimation. The system is ideal for applications in human resource analytics, workforce planning, and organizational compensation management.

The project involves a comprehensive pipeline starting from data collection and preprocessing to model training, evaluation, and deployment. Preprocessing techniques include handling missing values, encoding categorical data, and normalizing numerical features, ensuring the dataset is clean and consistent for modeling. Using Scikit-learn, the Linear Regression model was trained on a portion of the data and validated on a test set to assess its accuracy and reliability. Evaluation metrics such as Mean Squared Error (MSE) and R^2 Score confirmed the model's effectiveness in capturing relationships between features and predicting salaries.

A key feature of the system is its user-friendly web interface built with Streamlit. The interface allows users to input employee details and receive instant salary predictions. Streamlit's interactive capabilities ensure a seamless user experience, making the system accessible even to those without technical expertise. The system is scalable and capable of handling large datasets, ensuring its utility across a wide range of use cases, from small businesses to enterprise-level applications.

By automating salary prediction, the platform saves time, reduces errors, and ensures data-driven decision-making. It is not only a tool for estimating salaries but also a step towards modernizing compensation strategies in organizations. With its scalability, reliability, and focus on user experience, the **Salary Prediction System** demonstrates the potential of machine learning in addressing real-world challenges effectively.

REFERENCES

[1]. **https://pandas.pydata.org/docs/**

[2]. **https://numpy.org/doc/**

[3]. **https://scikit-learn.org/stable/**

[4]. *Géron, Aurélien.* Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. *O'Reilly Media, 2019.*

[5].https://docs.streamlit.io/

[6]. *Bishop, Christopher M.* ***Pattern Recognition and Machine Learning. Springer, 2006.***

[7]. *Deng, Li, and Dong Yu.* ***Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, 2014.***