SALARY PREDICTION SYSTEM

AI/ML Project report
submitted in partial fulfilment of the
requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

By

S.V.N Ramakanth

1602-19-733-118

Nikhil Rachha

1602-19-733-082

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2022

Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)

Hyderabad-500 031

Department of Computer Science & Engineering

DECLARATION BY THE CANDIDATE

I, **S.V.N Ramakanth,** bearing hall ticket number, **1602-19-733-118,** hereby declare that the project report entitled **Salary Prediction System**, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering**.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

SVN. Ramakanth

1602-19-733-118

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Computer Science & Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled **Salary Prediction System** being submitted by **S.V.N. Ramakanth,** bearing **1602-19-733-118,** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by her under my guidance.

Dr. T. Adilakshmi,

Professor & HOD,

Dept. of CSE

ABSTRACT

Predicting justified salary for an employee has always been a challenging job for an employer and so is for a new joinee. This Salary Prediction System eases the task as it uses the stack over flow 2021 dataset and helps one predict a person's salary based on his work location i.e. his country of residence, his education level and his experience in years. Instead of dividing the data set into testing and training sections, I've used all the data to train the model and then checked its error using the mean square error by predicting existing salary values based on the other attributes above mentioned and verified them with the actual values from the dataset.

ACKNOWLEDGEMENT

With immense pleasure, I record my deep sense of gratitude to our guide Komal Kaur, Professor, Vasavi College of Engineering, Hyderabad, for the keen interest and thorough encouragement extended throughout the period of the project work. This project would add as an asset to my academic profile.

TABLE OF CONTENTS

1. INTRODUCTION

“Prediction” refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be. Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be about all kinds of things and this Salary Prediction System is one such model.

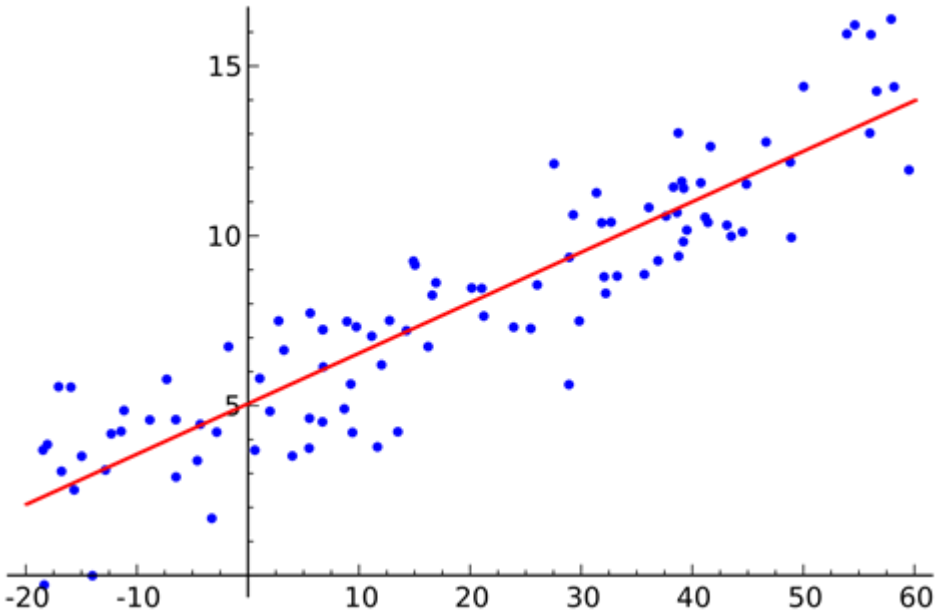
This model predicts a person’s salary based on his work location i.e. his country of residence, his education level and his experience in years. Firstly, I downloaded the stack over flow 2021 dataset that helps us train the model. Preprocessed the data based on requirements using Label Encoder and then instead of dividing the data set into testing and training sections, I’ve used all the data to train the model, I used various regression models from sklearn like Linear, Decision Tree and Random Forest Regression and then checked error for each one using the mean square error by predicting existing salary values based on the other attributes above mentioned and verified them with the actual values from the dataset. I then made a simple web page using the streamlit library to showcase the model.

2.Regression models evaluated:

1. Linear Regression
2. Decision Tree Regression
3. Random Forest Regression

2.1 LINEAR REGRESSION

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.



Linear regression using sklearn : [LinearRegression()]

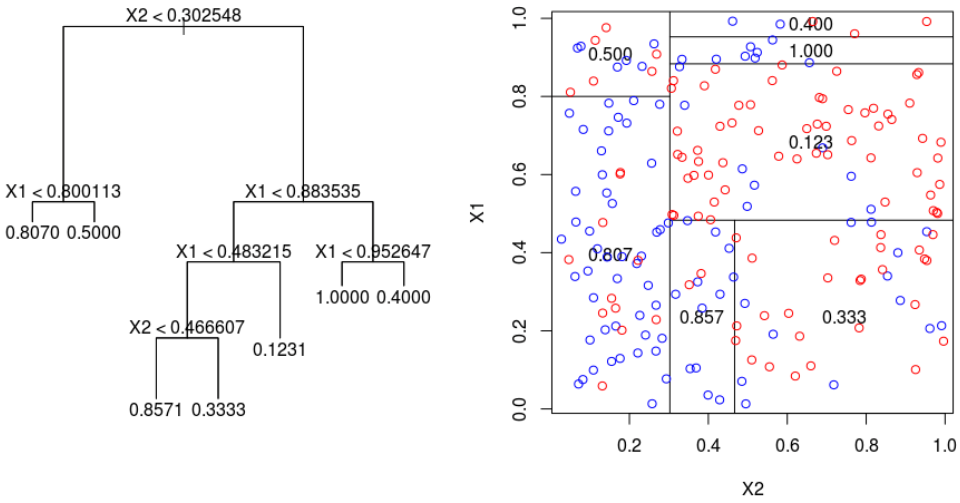
2.2 DECISION TREE REGRESSOR

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.

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

Decision Tree Regression:
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.



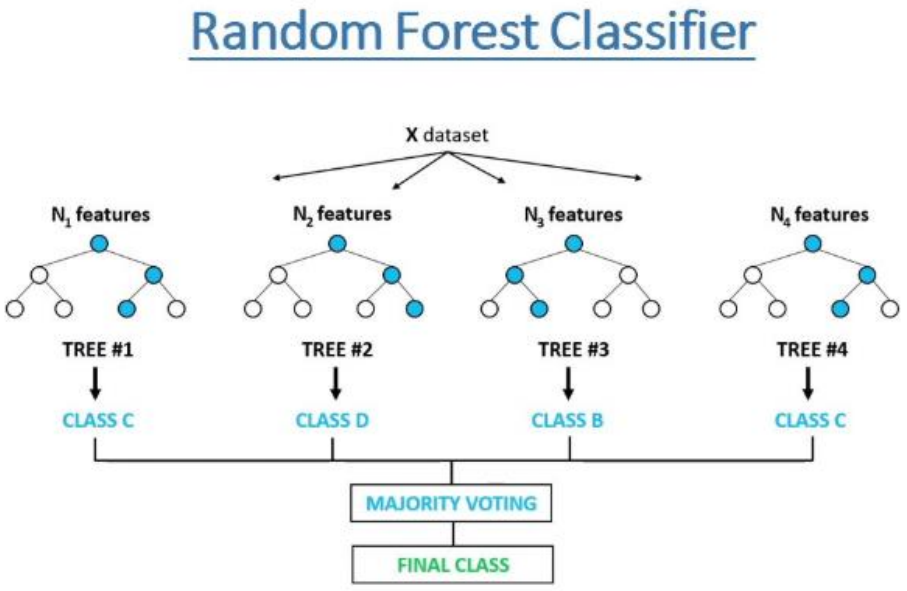
Decision Tree Regressor using sklearn : [DecisionTreeRegressor()]

2.3 RANDOM FOREST REGRESSOR

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 on 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 called Aggregation.

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 dataset forming sample datasets for every model. This part is called Bootstrap.



We need to approach the Random Forest regression technique like any other machine learning technique

- Design a specific question or data and get the source to determine the required data.
- Make sure the data is in an accessible format else convert it to the required format.
- Specify all noticeable anomalies and missing data points that may be required to achieve the required data.
- Create a machine learning model
- Set the baseline model that you want to achieve
- Train the data machine learning model.
- Provide an insight into the model with test data
- Now compare the performance metrics of both the test data and the predicted data from the model.
- If it doesn’t satisfy your expectations, you can try improving your model accordingly or dating your data, or using another data modeling technique.
- At this stage, you interpret the data you have gained and report accordingly.

Decision Tree Regressor using sklearn : RandomForestRegressor()

3. SYSTEM REQUIREMENTS

3.1 Hardware Requirements:

- Minimum RAM required: 512 MB
- Input devices: Mouse, Keyboard
- Output devices: Monitor

3.2 Software Requirements:

- Anaconda
- Python IDLE
- Numpy
- Pandas
- Matplotlib
- Sklearn
- Pickle
- Jupyter Notebook
- Streamlit etc.

4. IMPLEMENTATION

4.1 Prediction Model:

4.1.1 Dataset : [stackoverflowSurvey2021]

In [92]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("survey_results_public.csv")
```

In [93]:
```
df = df[["Country", "EdLevel", "YearsCodePro", "Employment", "ConvertedCompYearly"]]
df = df.rename({"ConvertedCompYearly": "Salary"}, axis=1)
df.head()
```

Out[93]:

	Country	EdLevel	YearsCodePro	Employment	Salary
0	Slovakia	Secondary school (e.g. American high school, G...	NaN	Independent contractor, freelancer, or self-em...	62268.0
1	Netherlands	Bachelor's degree (B.A., B.S., B.Eng., etc.)	NaN	Student, full-time	NaN
2	Russian Federation	Bachelor's degree (B.A., B.S., B.Eng., etc.)	NaN	Student, full-time	NaN
3	Austria	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	NaN	Employed full-time	NaN
4	United Kingdom of Great Britain and Northern I...	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	10	Independent contractor, freelancer, or self-em...	NaN

4.1.2 Preprocessing Data:

In [111]:
```
from sklearn.preprocessing import LabelEncoder
le_education = LabelEncoder()
df['EdLevel'] = le_education.fit_transform(df['EdLevel'])
df["EdLevel"].unique()
#le.classes_
```

Out[111]:
```
array([2, 0, 3, 1])
```

In [112]:
```
le_country = LabelEncoder()
df['Country'] = le_country.fit_transform(df['Country'])
df["Country"].unique()
```

Out[112]:
```
array([13, 12,  4, 15,  2,  3, 14, 16, 11,  6, 17,  1,  7,  8, 10,  0,  5,
       9])
```

4.1.3 Evaluating regression models and finding the best fit:

In [115]:
```
y_pred = linear_reg.predict(X)
```

In [116]:
```
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
error = np.sqrt(mean_squared_error(y, y_pred))
```

In [117]:
```
error
```

Out[117]:
```
41838.096750079996
```

In [118]:
```
from sklearn.tree import DecisionTreeRegressor
dec_tree_reg = DecisionTreeRegressor(random_state=0)
dec_tree_reg.fit(X, y.values)
```

Out[118]:
```
DecisionTreeRegressor(random_state=0)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [119]:
```
y_pred = dec_tree_reg.predict(X)
```

In [120]:
```
error = np.sqrt(mean_squared_error(y, y_pred))
print("${:,.02f}".format(error))
```

```
$31,267.18
```

In [121]:
```
from sklearn.ensemble import RandomForestRegressor
random_forest_reg = RandomForestRegressor(random_state=0)
random_forest_reg.fit(X, y.values)
```

Out[121]:
```
RandomForestRegressor(random_state=0)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

4.1.4 Dumping data into pickle files:

In [133]:
```
import pickle
```

In [134]:
```
data = {"model": regressor, "le_country": le_country, "le_education": le_education}
with open('saved_steps.pkl', 'wb') as file:
    pickle.dump(data, file)
```

In [135]:
```
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"]
```

4.2 Web app using streamlit library:

4.2.1 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()
```

4.2.2 predict_page.py:

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


def load_model():
    with open(r"C:\Users\USER\Desktop\SPS\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 of America",
    "India",
    "United Kingdom of Great Britain and Northern Ireland",
    "Germany",
    "Canada",
    "Brazil",
    "France",
    "Spain",
    "Australia",
    "Netherlands",
    "Poland",
    "Italy",
    "Russian Federation",
    "Sweden",
    "Turkey",
    "Switzerland",
    "Israel",
    "Norway",
)

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}")
```

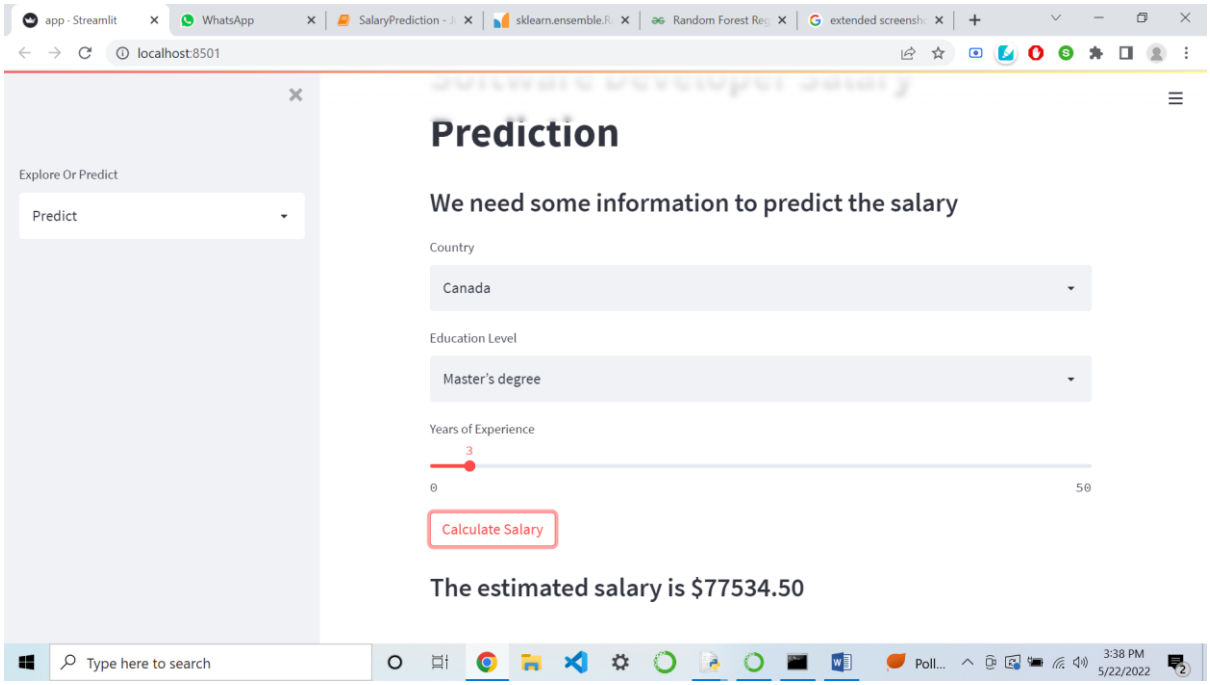
4.2.3 explore_page.py:

```
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
def load_data():
    df =
pd.read_csv(r"C:\Users\USER\Desktop\SPS\survey_results_public.cs
```

```
v")
    df = df[["Country", "EdLevel", "YearsCodePro", "Employment",
"ConvertedCompYearly"]]
    df = df[df["ConvertedCompYearly"].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["ConvertedCompYearly"] <= 250000]
    df = df[df["ConvertedCompYearly"] >= 10000]
    df = df[df["Country"] != "Other"]

    df["YearsCodePro"] =
df["YearsCodePro"].apply(clean_experience)
    df["EdLevel"] = df["EdLevel"].apply(clean_education)
    df = df.rename({"ConvertedCompYearly": "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=Tru
e)
    st.bar_chart(data)

    st.write(
        """
    #### Mean Salary Based On Experience
    """
    )

    data =
df.groupby(["YearsCodePro"])["Salary"].mean().sort_values(ascendin
g=True)
    st.line_chart(data)
```

5. RESULT


app - Streamlit
WhatsApp
localhost:8501
Explore Or Predict
Predict
Prediction
We need some information to predict the salary
Country
Canada
Education Level
Master's degree
Years of Experience
3
0
50
Calculate Salary
The estimated salary is $77534.50
Type here to search
3:38 PM
5/22/2022

Software Developer Salary Prediction

We need some information to predict the salary

Country

Canada

Education Level

Bachelor's degree

Years of Experience

3

0 50

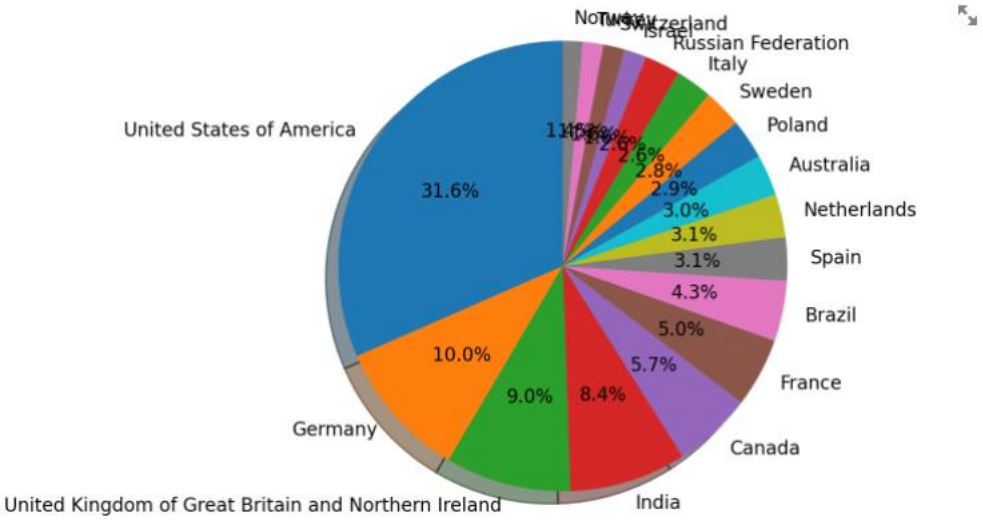
Calculate Salary

The estimated salary is $66585.07

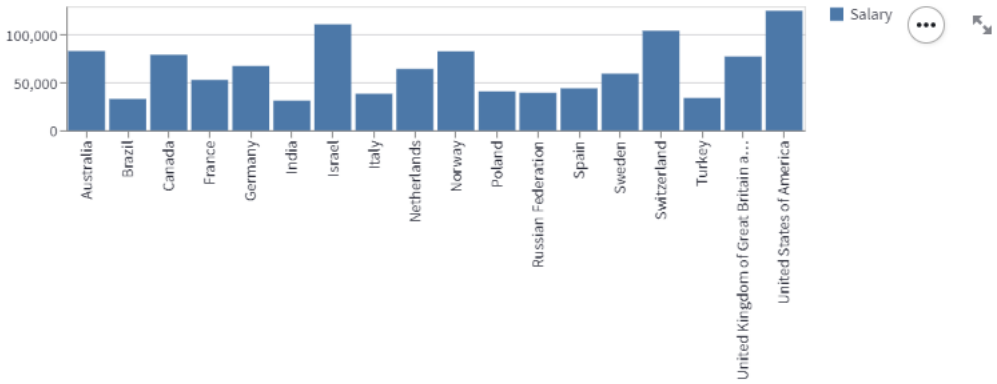
Explore Software Engineer Salaries

Stack Overflow Developer Survey 2020

Number of Data from different countries



Mean Salary Based On Country



6. REFERENCES

- https://scikit-learn.org/0.21/documentation.html
- https://docs.streamlit.io/