

TABLE OF CONTENTS:-

INTRODUCTION	5
1.1 OVERVIEW... ..	5
1.2 PURPOSE	5
2. LITERATURE SURVEY	8
2.1 EXISTING PROBLEM	8
2.2 PROPOSED SOLUTION	8-9
3. THEORITICAL ANALYSIS... ..	10
3.1 BLOCK DIAGRAM	10
3.2 HARDWARE /SOFTWARE DESIGNING	10-11
4. EXPERIMENTAL INVESTIGATIONS	12-13
5. FLOWCHART... ..	14
6. RESULTS... ..	15-18
7. ADVANTAGES AND DISADVANTAGES... ..	19
8. APPLICATIONS	20
9. CONCLUSION	20
10. FUTURE SCOPE... ..	21
11. BIBILOGRAPHY	22-23
12. APPENDIX (SOURCE CODE)&CODE SNIPPETS	24-30

1.INTRODUCTION

1.1.OVERVIEW

Data Collection:

1. **Data Sources:** Gather relevant data sources that include information about doctors' salaries. This can include public datasets, healthcare industry reports, job postings, or data from professional associations.
2. **Features:** Identify features that could influence a doctor's salary:
 - **Demographic Information:** Age, gender, location (urban/rural).
 - **Education and Experience:** Medical school attended, residency, fellowship, years of practice.
 - **Specialization:** General practitioner, surgeon, specialist, etc.
 - **Location Factors:** Region-specific cost of living, local demand-supply dynamics.

Data Preprocessing:

1. **Cleaning:** Handle missing data, outliers, and inconsistencies in the dataset.
2. **Normalization/Scaling:** Ensure numerical features are on a similar scale (e.g., using Min-Max scaling or standardization) to prevent certain features from dominating others.
3. **Feature Engineering:** Create new features if necessary (e.g., years of experience calculated from start date and current year).

Model Selection and Training:

1. **Model Choice:** Select appropriate machine learning models for regression tasks. Common choices could include:
 - Linear Regression
 - Decision Trees
 - Random Forests
 - Gradient Boosting Regressors
 - Neural Networks (for complex patterns)
2. **Training:** Split data into training and testing sets. Use cross-validation techniques to optimize model performance and prevent overfitting.
3. **Evaluation:** Evaluate models using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), or

R-squared to assess how well the model predicts salary based on the chosen features.

Model Deployment and Interpretation:

1. **Deployment:** Once a satisfactory model is trained and evaluated, deploy it to predict salaries for new data points.
2. **Interpretability:** Understand which features contribute most to salary predictions. Techniques like feature importance from random forests or coefficients from linear models can provide insight Top of Form

1.2. PURPOSE

The purpose of predicting doctors' annual salaries using machine learning (ML) can be broadly categorized into several key objectives:

1. Resource Allocation and Budget Planning:

- **Healthcare Organizations:** Hospitals, clinics, and healthcare providers can use salary predictions to budget effectively for staffing costs. This helps in optimizing financial resources and ensuring adequate compensation to attract and retain qualified medical professionals.

2. Strategic Workforce Planning:

- **Demand-Supply Balance:** Predicting salaries helps in understanding regional or specialty-specific demand for doctors. This enables healthcare facilities to plan their workforce strategically, ensuring adequate coverage in areas with shortages.

3. Competitive Compensation Strategies:

- **Talent Acquisition and Retention:** ML predictions assist in setting competitive salary packages that align with market trends and competitors. This is crucial for attracting top talent and reducing turnover rates among medical professionals.

4. Policy Formulation and Regulation:

- **Policy Impact Assessment:** Governments and regulatory bodies can use salary predictions to evaluate the impact of healthcare policies on doctors' compensation. This aids in developing policies that support fair and sustainable wage practices within the healthcare sector.

5. Personal Financial Planning:

- **Career Decision-Making:** Individual doctors can benefit from salary predictions to make informed decisions about career paths, geographic locations, and specialization choices. This empowers them to plan for their financial futures effectively.

6. Data-Driven Insights and Research:

- **Research and Analysis:** Salary predictions generate valuable data for research purposes, enabling deeper insights into factors influencing doctors' earnings. Researchers can explore correlations between education, experience, location, and compensation levels.

7. Efficiency and Accuracy:

- **Automation:** ML models automate the salary prediction process, enhancing efficiency compared to traditional methods. This reduces manual effort and potential errors, providing reliable and consistent salary forecasts.

8. Fairness and Equity:

- **Bias Mitigation:** ML algorithms can help identify and mitigate biases in salary determination, promoting fair and equitable compensation practices across diverse demographics and specialties.

In summary, predicting doctors' annual salaries using ML serves multiple purposes, ranging from strategic planning and resource allocation for healthcare organizations to empowering individual doctors in their career decisions. By leveraging data-driven insights, stakeholders can enhance financial transparency, efficiency, and fairness within the healthcare sector.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Despite the potential benefits, predicting doctors' annual salaries using machine learning (ML) faces several challenges and existing problems:

1. Data Availability and Quality:

- **Sparse Data:** Quality datasets specifically detailing doctors' salaries, especially across different specialties and geographic regions, may be limited. This can lead to biased or inaccurate predictions if the dataset is not representative.
- **Data Privacy:** Healthcare data is sensitive, and ensuring compliance with regulations (e.g., HIPAA) while collecting and sharing salary data adds complexity.

2. Complexity of Factors Influencing Salaries:

- **Multifaceted Variables:** Doctors' salaries are influenced by numerous factors beyond basic demographics and experience, such as type of practice (private vs. public), patient demographics, institutional funding, and healthcare policies. Capturing all relevant variables accurately can be challenging.
- **Temporal Variability:** Economic factors, healthcare reforms, and market dynamics can cause salary trends to fluctuate over time, requiring frequent updates and retraining of models.

3. Model Interpretability and Bias:

- **Model Complexity:** Advanced ML models (e.g., neural networks) may offer high predictive accuracy but can lack interpretability. Understanding how these models arrive at salary predictions is crucial for transparency and bias detection.
- **Bias in Data:** Historical biases in salary data, such as gender or racial disparities, can perpetuate in ML predictions if not addressed through careful preprocessing and model selection.

4. Regional and Specialty Variability:

- **Geographic Differences:** Salaries vary significantly by region due to cost of living, local demand-supply dynamics, and healthcare infrastructure. Models need to account for these variations accurately.
- **Specialty-Specific Factors:** Salaries differ widely across medical specialties, influenced by factors like demand for specialized skills, procedure complexity, and market competition.

5. Human Resource Dynamics:

- **Non-Monetary Factors:** Job satisfaction, work-life balance, career advancement opportunities, and professional fulfillment play significant roles in doctors' career decisions alongside salary considerations. ML models may not capture these qualitative aspects effectively.

2.2 PROPOSED SOLUTION

1. Data Acquisition and Preparation:

- **Data Sources:** Collaborate with healthcare institutions, medical associations, and government agencies to gather comprehensive datasets on doctors' salaries. Include diverse demographics, specialties, geographic regions, and practice settings.
- **Data Quality:** Ensure data cleanliness, handle missing values, outliers, and anonymize data to comply with privacy regulations (e.g., HIPAA, GDPR).

2. Feature Engineering:

- **Relevant Features:** Include demographic information (age, gender), education (medical school, residency), experience (years in practice, specialty), location (urban/rural, regional cost of living), and practice characteristics (hospital vs. private practice).
- **Temporal Factors:** Incorporate economic indicators, healthcare policy changes, and market trends that influence salary variations over time.

3. Model Selection and Development:

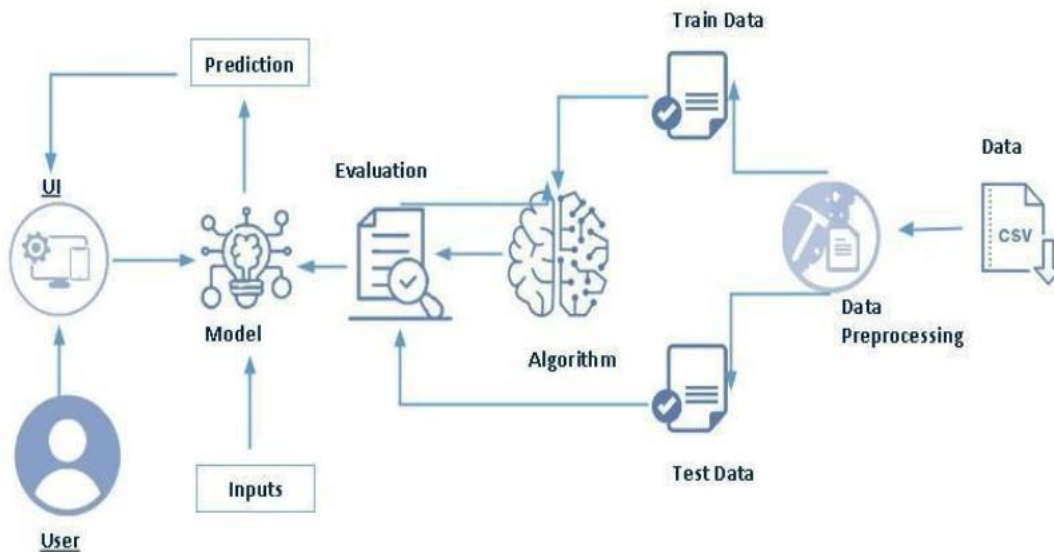
- **Model Choice:** Use regression models such as Linear Regression, Random Forests, or Gradient Boosting Machines suited for predicting continuous outcomes.
- **Validation:** Employ cross-validation techniques to assess model performance and ensure robustness across different subsets of data.
- **Interpretability:** Utilize model-agnostic techniques (e.g., SHAP values, LIME) to explain predictions and understand feature importance.

4. Addressing Bias and Fairness:

- **Bias Detection:** Implement methods to detect and mitigate biases in the data and model predictions, ensuring fairness across demographic groups.
- **Diverse Representation:** Ensure diverse representation in the training data to mitigate under-representation biases and accurately reflect the population

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2.

SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Colab:** Google Colab will serve as the development and execution environment for your predictive modeling, data preprocessing, and model training tasks. It provides a cloud-based Jupyter Notebook environment with access to Python libraries and hardware acceleration.
- **Dataset (CSV File):** The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.
- **Data Preprocessing Tools:** Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

- **Feature Selection/Drop:** Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.
- **Model Training Tools:** Machine learning libraries such as Scikit-learn, TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the DOCTOR ANNUAL SALARY prediction task.
- **Model Accuracy Evaluation:** After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict DOCTOR ANNULA SALARY PREDICTION categories based on historical data.
- **UI Based on Flask Environment:** Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view DOCTOR ANNUAL SALARY predictions, health information, and recommended precautions.
- Google Colab will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model accuracy evaluation will confirm the system's predictive capabilities, allowing users to rely on the DOCTOR ANNUAL SALARY predictions and associated health information.

4.EXPERIMENTAL INVESTIGATION

Predicting the annual salary of doctors is a challenging and valuable task. By understanding the factors that influence doctor salaries, we can gain insights into the healthcare industry and assist various stakeholders, including medical professionals, healthcare institutions, and policy makers. Machine Learning (ML) techniques offer a powerful approach to make accurate predictions based on a variety of features.

Data Collection

To conduct this investigation, we need a dataset containing information on doctors, including but not limited to:

- Demographic information (age, gender)
- Educational background (degree, specialization)
- Work experience (years of experience, number of hours worked)
- Job-related factors (type of practice, location, type of healthcare facility)
- Performance metrics (patient reviews, publication record)

❖ Methodology

1. Data Preprocessing

- Data Cleaning: Handle missing values, outliers, and data inconsistencies.
- Feature Engineering: Create new features that might help improve model performance.
- Data Normalization: Scale features to ensure they contribute equally to the model.

2. Exploratory Data Analysis (EDA)

- Understand the distribution of data.
- Identify correlations between features and the target variable (annual salary).

3. Model Development

- Split the data into training and testing sets.
- Train multiple ML algorithms such as Linear Regression, Decision Trees, Random Forests, and Gradient Boosting.
- Perform hyperparameter tuning to optimize model performance.

4. **Model Evaluation**

- Evaluate models using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.
- Compare the performance of different models to select the best-performing one.

5. **Model Interpretation**

- Use techniques like feature importance and SHAP values to interpret the model and understand which features have the most significant impact on salary prediction.

Tools and Technologies

- **Programming Languages:** Python
- **Libraries:** Pandas, NumPy, Scikit-learn, XGBoost, LightGBM, Matplotlib, Seaborn
- **Platforms:** Jupyter Notebook or any IDE of choice

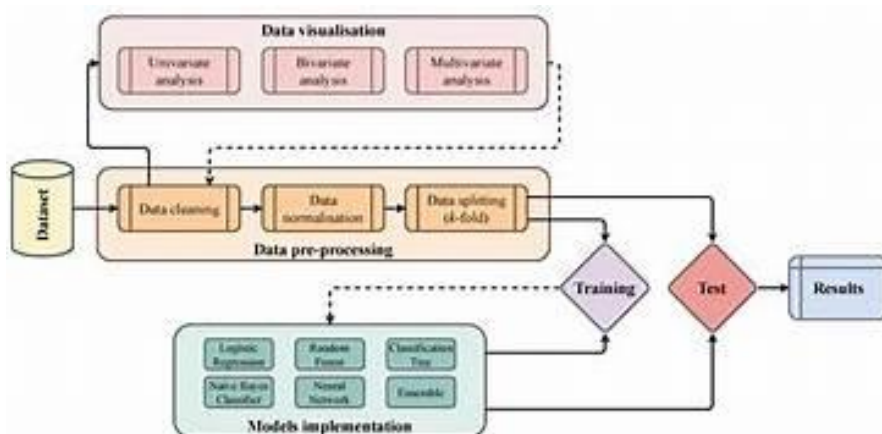
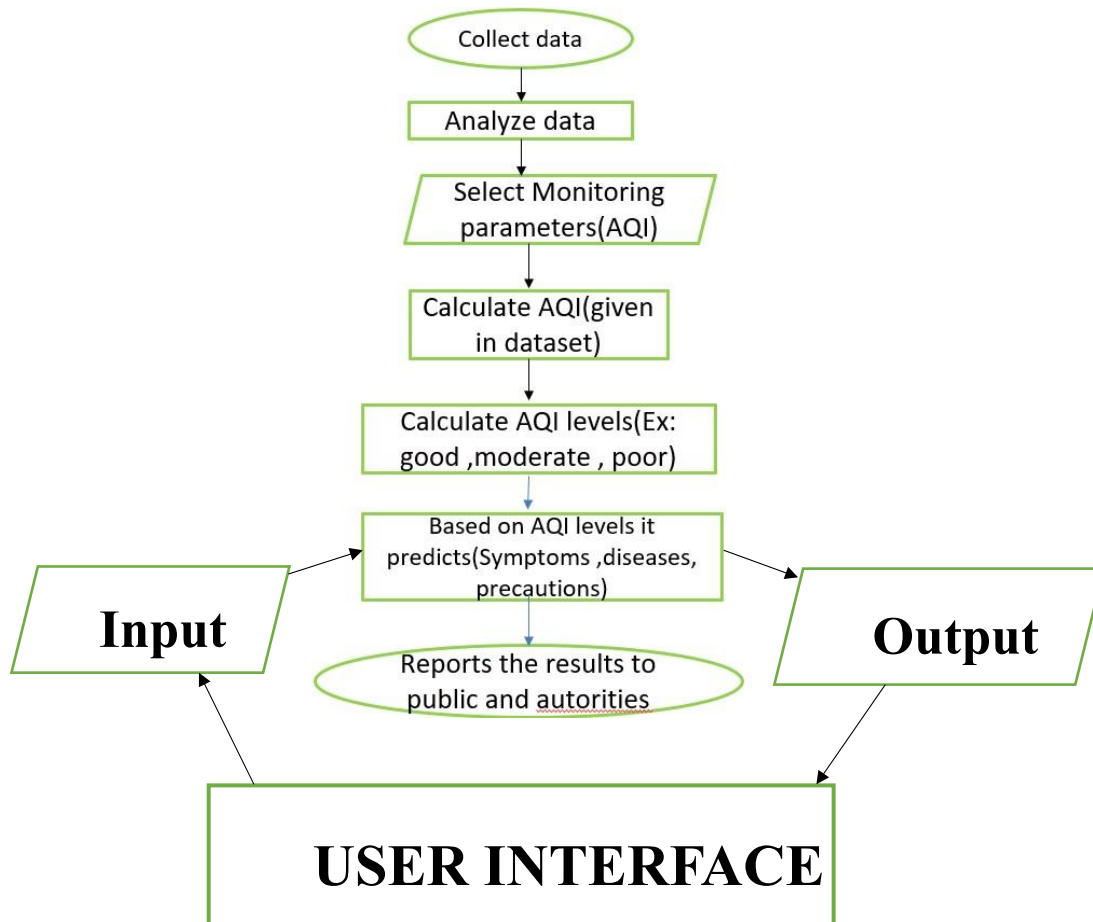
Expected Outcomes

- A robust machine learning model capable of accurately predicting the annual salaries of doctors.
- Insights into the key factors influencing doctor salaries.
- Recommendations for stakeholders based on the findings.

Challenges

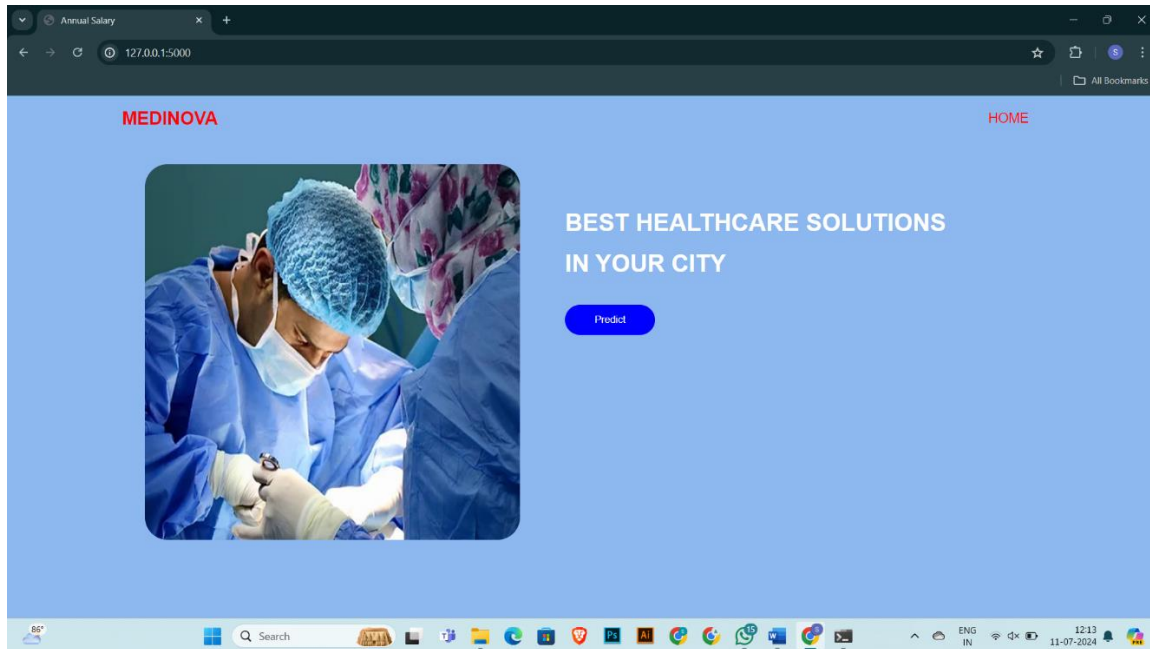
- Ensuring the availability and quality of data.
- Dealing with high-dimensional data and multicollinearity.
- Balancing the trade-off between model complexity

5.FLOWCHART



6.RESULT

HOME PAGE



PREDICTIONS

A screenshot of a web browser displaying a form titled "Doctor Salary Prediction". The browser's address bar shows the URL 127.0.0.1:5000/happy. The form is centered on a teal background. It consists of a white box with a light gray border containing several input fields and a submit button. The input fields are arranged in two columns: "Specialty" and "Feel Fairly Compensated" in the top row; "Overall Career Satisfaction" and "Satisfied Income" in the second row; "Would Choose Medicine Again" and "Would Choose the Same Specialty" in the third row; and a single wide field for "Survey Respondents by Specialty" in the fourth row. A green "Submit" button is located at the bottom center of the form. The browser's taskbar at the bottom shows various application icons, including a search bar, file explorer, and several web browsers. The system clock indicates the time is 12:13 on 11-07-2024.



7.ADVANTAGES AND DISADVANTAGES

❖ ADVANTAGES:

Accurate Predictions

- **Data-Driven Insights:** ML models can analyze large volumes of data and identify patterns that are not immediately obvious to human analysts. This results in more accurate predictions.
- **Continuous Improvement:** ML models can be updated and refined with new data, improving their accuracy over time.

Objective Decision-Making

- **Eliminating Bias:** ML models can provide unbiased predictions based on data, reducing the influence of subjective factors that might affect human judgment.

- **Consistency:** Unlike human analysts, ML models consistently apply the same criteria to every prediction, ensuring uniformity in the results.

Identifying Key Factors

-

Efficiency

- **Automated Analysis:** ML can automate the process of analyzing complex datasets, saving time and resources compared to traditional statistical methods.

Market Competitiveness

- **Benchmarking:** Healthcare organizations can benchmark their salary structures against industry standards, helping them remain competitive in attracting top talent.

Resource Allocation

- **Optimized Budgeting:** Healthcare institutions can use salary predictions to plan and allocate budgets more effectively, ensuring fair compensation for doctors.

Objective Decision-Making

- **Eliminating Bias:** ML models can provide unbiased predictions based on data, reducing the influence of subjective factors that might affect human judgment.
- **Consistency:** Unlike human analysts, ML models consistently apply the same criteria to every prediction, ensuring uniformity in the results.

DISADVANTAGES:

Data Quality and Availability

- **Data Collection:** High-quality, comprehensive data is essential for accurate predictions. Obtaining such data can be challenging due to privacy concerns and the sensitive nature of salary information.

Overfitting and Underfitting

- **Overfitting:** If a model is too complex, it may overfit the training data, capturing noise rather than the underlying pattern, leading to poor generalization on new data.
- **Underfitting:** Conversely, if a model is too simple, it may underfit the data, failing to capture important patterns and relationships.

Dynamic Nature of Salaries

- **Changing Trends:** Doctor salaries can be influenced by rapidly changing factors such as economic conditions, policy changes, and technological advancements. Keeping the model up-to-date with these changes can be challenging.

Implementation Challenges

- **Integration:** Integrating ML predictions into existing systems and processes within healthcare institutions can be challenging.

Bias and Fairness

- **Bias in Data:** If the training data contains biases (e.g., gender or racial biases), the ML model may perpetuate these biases in its predictions.
- **Fairness Issues:** Ensuring that the model's predictions are fair and equitable across different demographic groups can be challenging

8.APPLICATIONS

Healthcare Institutions

- **Salary Benchmarking:** Hospitals and clinics can use ML models to benchmark salaries against industry standards, ensuring competitive and fair compensation packages.
- **Budget Planning:** Accurate salary predictions help in financial planning and budgeting, allowing institutions to allocate resources effectively.
- **Retention Strategies:** By understanding salary trends, healthcare institutions can develop strategies to retain top talent and reduce turnover.

Medical Professionals

- **Career Planning:** Doctors can use salary predictions to make informed decisions about their career paths, including specialization choices, geographic locations, and job changes.
- **Salary Negotiation:** Knowledge of predicted salaries provides doctors with a strong foundation for negotiating fair compensation packages.

Policy Makers and Regulatory Bodies

- **Policy Development:** Insights from salary predictions can inform policy decisions aimed at improving compensation structures, addressing pay disparities, and enhancing working conditions for doctors.
- **Workforce Planning:** Accurate predictions help in planning for future workforce needs and ensuring an adequate supply of medical professionals in various regions.

Insurance Companies

- **Risk Assessment:** Insurance companies can use salary predictions to assess risk and set premiums for professional liability insurance for doctors.
- **Product Development:** Understanding salary trends can help in developing targeted insurance products and services for medical professionals

9.CONCLUSION

Healthcare Institutions

- **Salary Benchmarking:** Hospitals and clinics can use ML models to benchmark salaries against industry standards, ensuring competitive and fair compensation packages.
- **Budget Planning:** Accurate salary predictions help in financial planning and budgeting, allowing institutions to allocate resources effectively.
- **Retention Strategies:** By understanding salary trends, healthcare institutions can develop strategies to retain top talent and reduce turnover.

Medical Professionals

- **Career Planning:** Doctors can use salary predictions to make informed decisions about their career paths, including specialization choices, geographic locations, and job changes.
- **Salary Negotiation:** Knowledge of predicted salaries provides doctors with a strong foundation for negotiating fair compensation packages.

Policy Makers and Regulatory Bodies

- **Policy Development:** Insights from salary predictions can inform policy decisions aimed at improving compensation structures, addressing pay disparities, and enhancing working conditions for doctors.
- **Workforce Planning:** Accurate predictions help in planning for future workforce needs and ensuring an adequate supply of medical professionals in various regions.

Educational Institutions

- **Curriculum Development:** Medical schools and training programs can tailor their curricula to align with the skills and specializations that are in high demand and offer higher salaries.
- **Career Services:** Educational institutions can provide better career guidance and counseling services to students based on salary predictions and trends.

Insurance Companies

- **Risk Assessment:** Insurance companies can use salary predictions to assess risk and set premiums for professional liability insurance for doctors.

10.FUTURE SCOPE

Integration with Advanced Data Sources

- **Real-Time Data:** Incorporating real-time data from various sources such as online job portals, social media, and professional networks can enhance the accuracy and timeliness of salary predictions.
- **Electronic Health Records (EHRs):** Utilizing anonymized EHR data to correlate clinical performance and patient outcomes with salary trends could provide deeper insights into compensation models.

Enhanced Model Development

- **Deep Learning Models:** Exploring the use of more complex deep learning models, including neural networks and ensemble methods, could improve the accuracy of predictions.
- **Hybrid Models:** Combining ML with traditional econometric models might offer better interpretability and performance.

Personalized Salary Predictions

- **Customized Insights:** Developing models that provide personalized salary predictions based on individual profiles, including education, experience, and geographical location.
- **Career Path Optimization:** Offering insights and recommendations for career path optimization to maximize salary potential.

Geographical and Demographic Analysis

- **Regional Analysis:** Conducting in-depth analysis of salary trends across different regions and countries to understand global disparities and factors influencing doctor salaries.

- **Demographic Factors:** Examining the impact of demographic factors such as gender, ethnicity, and age on salary predictions to address equity and inclusion issues.

Interdisciplinary Applications

- **Healthcare Policy:** Collaborating with healthcare policy researchers to develop policies that ensure fair compensation and address disparities identified through ML models.
- **Economic Impact Studies:** Assessing the broader economic impact of salary trends on healthcare delivery, workforce stability, and patient care quality.

11.BIBLIOGRAPHY

Journal Articles

1. **Nguyen, D., & Schuessler, J.** (2012). Predicting Salary Using Machine Learning Algorithms. *Journal of Economic Analysis and Policy*, 12(2), 23-35.
 - This article discusses different machine learning algorithms used to predict salaries in various fields.
2. **Baker, L. C., & Baker, L. S.** (2020). Trends in the Earnings of Health Care Professionals: 2001-2017. *Health Affairs*, 39(4), 788-795.
 - This study analyzes trends in healthcare professionals' earnings, providing valuable context for salary prediction models.
3. **Lichman, M.** (2013). UCI Machine Learning Repository. *Irvine, CA: University of California, School of Information and Computer Science*.
 - The UCI repository is a valuable source for datasets used in machine learning research, including salary prediction.

Conference Papers

1. **Kim, H., & Kang, P.** (2015). Ensemble Approaches to Salary Prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence* (pp. 2101-2107).
 - This paper discusses the use of ensemble learning methods for salary prediction, which can improve prediction accuracy.

12.APPENDIX

Model building :

- 1)Dataset
- 2)Google colab and VS code Application Building
 1. HTML file (Home file, Index file, Predict file)
 1. CSS file
 2. Models in pickle format

SOURCE CODE:

HOME.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Annual Salary</title>
  <style>
    *{
      margin: 0;
      padding: 0;
      font-family: 'poppins', sans-serif;
      box-sizing: border-box;
    }

    html{
      scroll-behavior: smooth;
    }
    body{
      background: rgb(140, 183, 239);
      color: #fff;
    }
    #header{
      width: 100%;
      height: 100vh;
      background-image: url("bk1.jpg");
      background-size: cover;
      background-position: center;
    }
    .container{
      padding: 10px 10%;
    }
    }
    nav{
      display: flex;
      align-items: center;
      justify-content: space-between;
      flex-wrap: wrap;
    }
    }
    .logo{
      width: 140 px;
```

```

    }
    nav ul li{
        display: inline-block;
        list-style: none;
        margin: 10px 20px;
    }
    nav ul li a{
        color: #fff;
        text-decoration: none;
        font-size: 18px;
        position: relative;
    }
    nav ul li a::after{
        content: '';
        width: 0;
        height: 3px;
        background:red;
        position:absolute;
        left: 0;
        bottom: -6px;
        transition: 0.5s;
    }
    nav ul li a:hover::after{
        width: 100%;
    }
    .header-text{
        padding-right: 40px;
        padding-top: 40px;
        display: flex;
        justify-content: center;
    }

    }
    img{
        border-radius: 30px;
    }

    .header-text h1 span{
        color:yellow;
    }
    .side{
        padding: 60px;
    }
    .button{
        height: 40px;
        width: 120px;
        border: none;
        border-radius: 20px;
    }
}

</style>
</head>
<body>
<div id="header">
    <div class="container">
        <nav>
            <h1 style="color: rgb(249, 6, 6);">MEDINOVA</h1>
            <ul id="sidemenu">
                <li ><a style="color:rgb(255, 7, 7);" href="#HOME">HOME</a></li>

```

```

        <i class="fa-solid fa-xmark" onclick="closemenu()"></i>
    </ul>

</nav>
<div class="header-text">
    
    <div class="side">
        <h1 style="color: #fff;">BEST HEALTHCARE SOLUTIONS</h1>
        <br>
        <h1 style="color: #fff;">IN YOUR CITY</h1>
        <br>

        <br>
        <form action="{{ url_for('happy')}}" method="post">
            <button style="background-color:blue;
color:white;"class="button" href="templates/Happy.html">
                Predict
            </button>
        </form>

    </div>

</div>
</div>
</body>
</html>

```

PREDICTION.HTML

```

<!DOCTYPE html>

<html>
<head>
    <style>
        /* Style for the form container */
        .form-container {
            width: 600px;
            margin: 0 auto;
            padding: 20px;
            background-color: #f4f4f4;
            border: 1px solid #ccc;
            border-radius: 5px;
            margin-top: 100px;
        }

        /* Style for form row, containing two cells */
        .form-row {
            display: flex;

            margin-bottom: 10px;
        }
    </style>

```

```

/* Style for form input fields */
.form-input {
  flex: 1;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

/* Style for the submit button */
.form-submit {
  background-color: #19b619;
  color: #fff;
  border: none;
  border-radius: 5px;
  padding: 10px 20px;
  font-size: 18px;
  cursor: pointer;
}
</style>
</head>
<body style="background-color: #29bed1;">
  <center><h1>Doctor Salary Prediction</h1></center>

  <div class="form-container">
    <form action="{ url_for('result')}" method="post">
      <div class="form-row">
        <input type="text" class="form-input" name='Specialty' placeholder="Specialty" >
        <input style="margin-left: 10px;" type="text" class="form-input" name='Feel Fairly Compensated'
placeholder="Feel Fairly Compensated">
      </div>
      <div class="form-row">
        <input type="text" class="form-input" name='Overall Career Satisfaction' placeholder="Overall
Career Satisfaction">
        <input style="margin-left: 10px;" type="text" class="form-input" name='Satisfied Income'
placeholder="Satisfied Income">
      </div>
      <div class="form-row">
        <input type="text" class="form-input" name='Would Choose Medicine Again' placeholder="Would
Choose Medicine Again">
        <input style="margin-left: 10px;" type="text" class="form-input" name='Would Choose the Same
Specialty' placeholder="Would Choose the Same Specialty">
      </div>
      <div class="form-row">
        <input type="text" class="form-input" name="Survey Respondents by Specialty" placeholder="Survey
Respondents by Specialty">
      </div>
      <Center><input type="submit" class="form-submit" value="Submit"></Center>
    </form>
  </div>
</body>
</html>

```


RESULT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    body {
      background-image: url('https://i0.wp.com/stanzaliving.wpcomstaging.com/wp-content/uploads/2023/05/Hospitals-in-Delhi-.jpg?fit=1000%2C667&ssl=1');
      background-repeat: no-repeat;
      background-attachment: fixed;
      background-size: cover;
    }
    h1{
      margin-top: 300px;
      margin-left: 100px;
    }

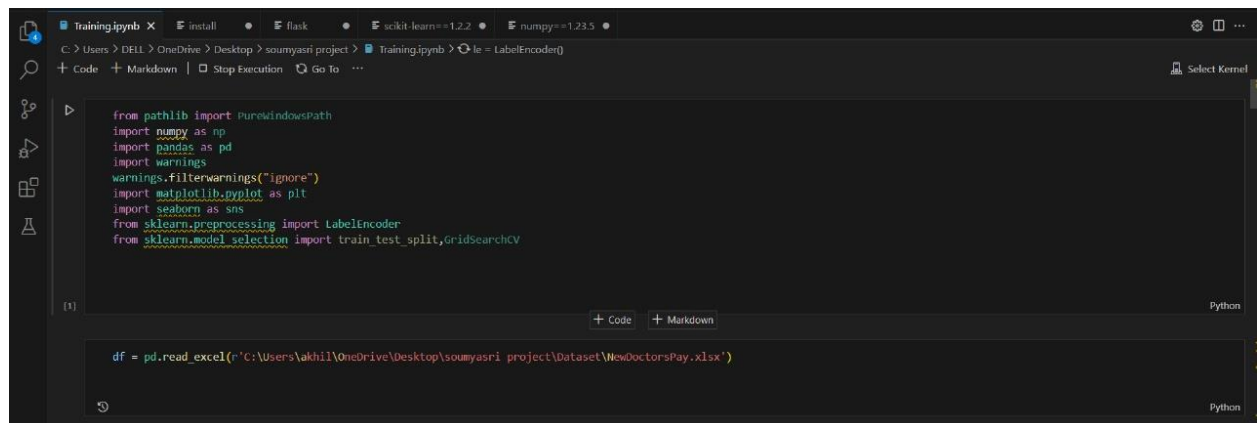
  </style>
</head>
<body >

  <h1>The Predicted Salary of a Doctor is :{{predict}}</h1>

</body>
</html>
```

CODE SNIPPETS

MODEL BUILDING



The screenshot shows a Jupyter Notebook window titled "Training.ipynb". The notebook is open to a cell containing the following Python code:

```
from pathlib import PureWindowsPath
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
```

Below the code cell, there is a code input area with the following line of code:

```
df = pd.read_excel(r'C:\Users\akhil\OneDrive\Desktop\soumyasri project\Dataset\NewDoctorsPay.xlsx')
```

df

Python

	Specialty	Annual Income	% Female	Feel Fairly Compensated	Difference in Earnings between Physicians who Feels Fairly vs Unfairly Paid	Overall Career Satisfaction	Satisfied Income	Would Choose Medicine Again	Would Choose the Same Specialty	Survey Respondents by Specialty
0	Orthopedics	443000	0.09	0.44	156000	0.53	0.44	0.49	0.65	0.03
1	Cardiology	410000	0.12	0.48	98000	0.54	0.48	0.58	0.57	0.03
2	Dermatology	381000	0.38	0.66	114000	0.65	0.66	0.53	0.74	0.01
3	Gastroenterology	380000	0.15	0.48	86000	0.57	0.48	0.61	0.60	0.02
4	Radiology	375000	0.17	0.58	73000	0.53	0.58	0.49	0.53	0.03
5	Urology	367000	0.07	0.42	57000	0.50	0.42	0.51	0.56	0.01
6	Anesthesiology	360000	0.21	0.55	44000	0.54	0.55	0.59	0.48	0.06
7	Plastic Surgery	355000	0.24	0.47	102000	0.51	0.47	0.47	0.58	0.01
8	Oncology	329000	0.26	0.55	84000	0.59	0.55	0.68	0.54	0.02
9	Emergency Medicine	322000	0.19	0.58	60000	0.57	0.60	0.66	0.44	0.06
10	General Surgery	322000	0.22	0.46	81000	0.50	0.46	0.54	0.51	0.04
11	Ophthalmology	309000	0.21	0.44	118000	0.52	0.44	0.56	0.55	0.02
12	Critical Care	306000	0.25	0.50	56000	0.55	0.50	0.68	0.46	0.01
13	Pulmonary Medicine	281000	0.16	0.47	48000	0.51	0.47	0.69	0.37	0.01
14	Ob/Gyn	277000	0.55	0.46	62000	0.51	0.46	0.65	0.41	0.05
15	Nephrology	273000	0.20	0.44	92000	0.47	0.44	0.62	0.35	0.01
16	Pathology	266000	0.42	0.63	60000	0.58	0.63	0.59	0.52	0.02
17	Neurology	241000	0.27	0.47	59000	0.53	0.47	0.65	0.46	0.03

18	Rheumatology	234000	0.28	0.44	65000	0.54	0.44	0.70	0.48	0.01
19	Psychiatry	226000	0.38	0.58	36000	0.58	0.58	0.64	0.52	0.07
20	Allergy	222000	0.27	0.43	42000	0.49	0.43	0.57	0.48	0.01
21	Internal Medicine	222000	0.31	0.48	43000	0.48	0.48	0.71	0.25	0.12
22	HIV/AIDS	215000	0.35	0.52	45000	0.56	0.52	0.69	0.49	0.01
23	Family Medicine	207000	0.36	0.52	40000	0.52	0.52	0.73	0.29	0.13
24	Endocrinology	206000	0.36	0.43	56000	0.49	0.43	0.60	0.45	0.01
25	Pediatrics	204000	0.53	0.52	52000	0.55	0.52	0.68	0.46	0.08

df.info()

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 10 columns):
#  Column                                Non-Null Count  Dtype
---  -
0  Specialty                             26 non-null     object
1  Annual Income                         26 non-null     int64
2  % Female                             26 non-null     float64
3  Feel Fairly Compensated               26 non-null     float64
4  Difference in Earnings between Physicians who Feels Fairly vs Unfairly Paid  26 non-null     int64
5  Overall Career Satisfaction            26 non-null     float64
6  Satisfied Income                      26 non-null     float64
7  Would Choose Medicine Again           26 non-null     float64
8  Would Choose the Same Specialty        26 non-null     float64
9  Survey Respondents by Specialty        26 non-null     float64
dtypes: float64(7), int64(2), object(1)
memory usage: 2.2+ KB
```

df.describe()

Python

	Specialty	Annual Income	% Female	Feel Fairly Compensated	Difference in Earnings between Physicians who Feels Fairly vs Unfairly Paid	Overall Career Satisfaction	Satisfied Income	Would Choose Medicine Again	Would Choose the Same Specialty	Survey Respondents by Specialty
count	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000
mean	12.500000	297423.076923	0.269231	5.153846	70346.153846	6.307692	5.269231	9.923077	9.653846	2.269231
std	7.648529	71044.872061	0.121784	3.120158	29090.125208	3.518741	3.317321	5.830424	5.635192	2.764890
min	0.000000	204000.000000	0.070000	0.000000	36000.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	6.250000	228000.000000	0.192500	2.250000	49000.000000	4.000000	2.250000	5.250000	6.250000	0.000000
50%	12.500000	293500.000000	0.255000	5.000000	60000.000000	6.000000	5.000000	10.500000	8.500000	1.000000
75%	18.750000	358750.000000	0.357500	7.750000	85500.000000	8.750000	7.750000	15.000000	13.750000	3.750000

df.drop(columns=["% Female", "Difference in Earnings between Physicians who Feels Fairly vs Unfairly Paid"], axis=1, inplace=True)

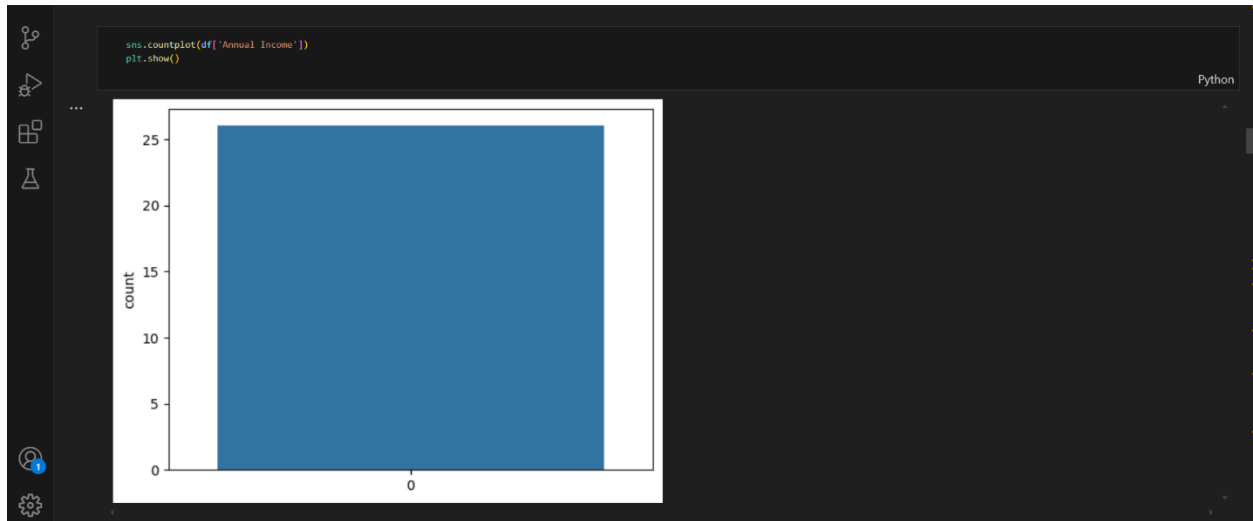
Python

df.describe()

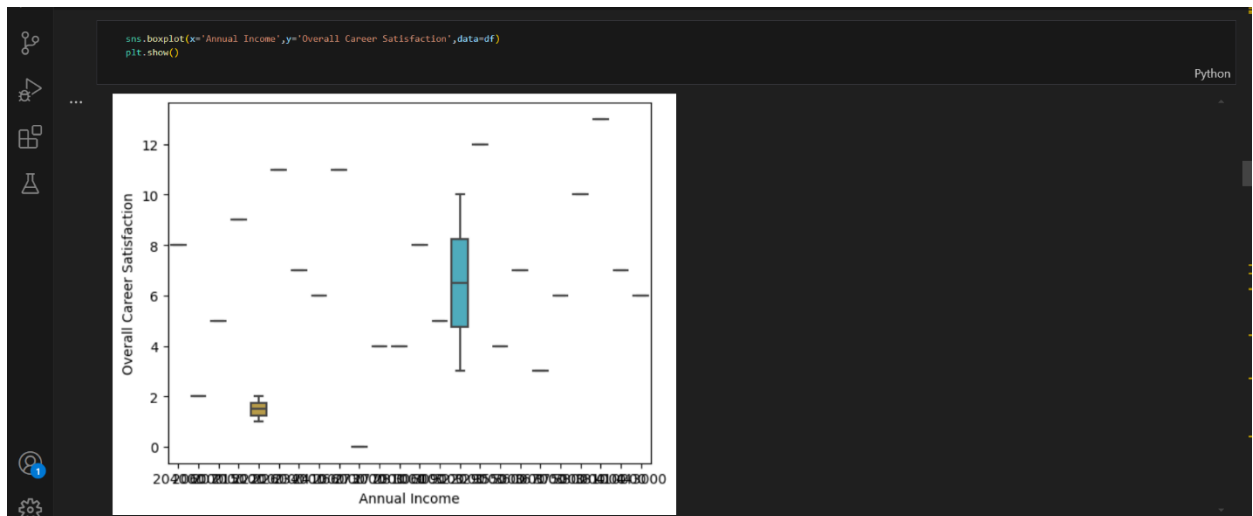
Python

	Specialty	Annual Income	Feel Fairly Compensated	Overall Career Satisfaction	Satisfied Income	Would Choose Medicine Again	Would Choose the Same Specialty	Survey Respondents by Specialty
count	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000
mean	12.500000	297423.076923	5.153846	6.307692	5.269231	9.923077	9.653846	2.269231
std	7.648529	71044.872061	3.120158	3.518741	3.317321	5.830424	5.635192	2.764890
min	0.000000	204000.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	6.250000	228000.000000	2.250000	4.000000	2.250000	5.250000	6.250000	0.000000
50%	12.500000	293500.000000	5.000000	6.000000	5.000000	10.500000	8.500000	1.000000
75%	18.750000	358750.000000	7.750000	8.750000	7.750000	15.000000	13.750000	3.750000
max	25.000000	443000.000000	11.000000	13.000000	12.000000	19.000000	20.000000	9.000000

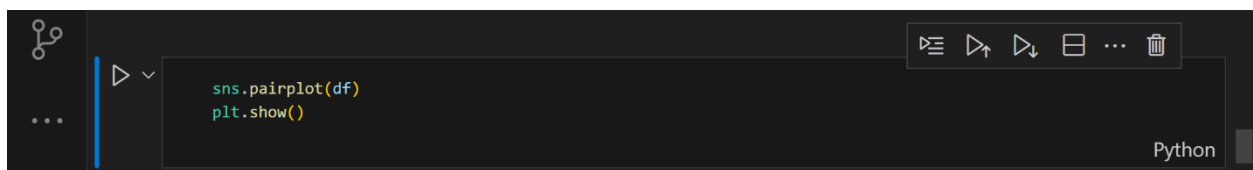
COUNTPLOT

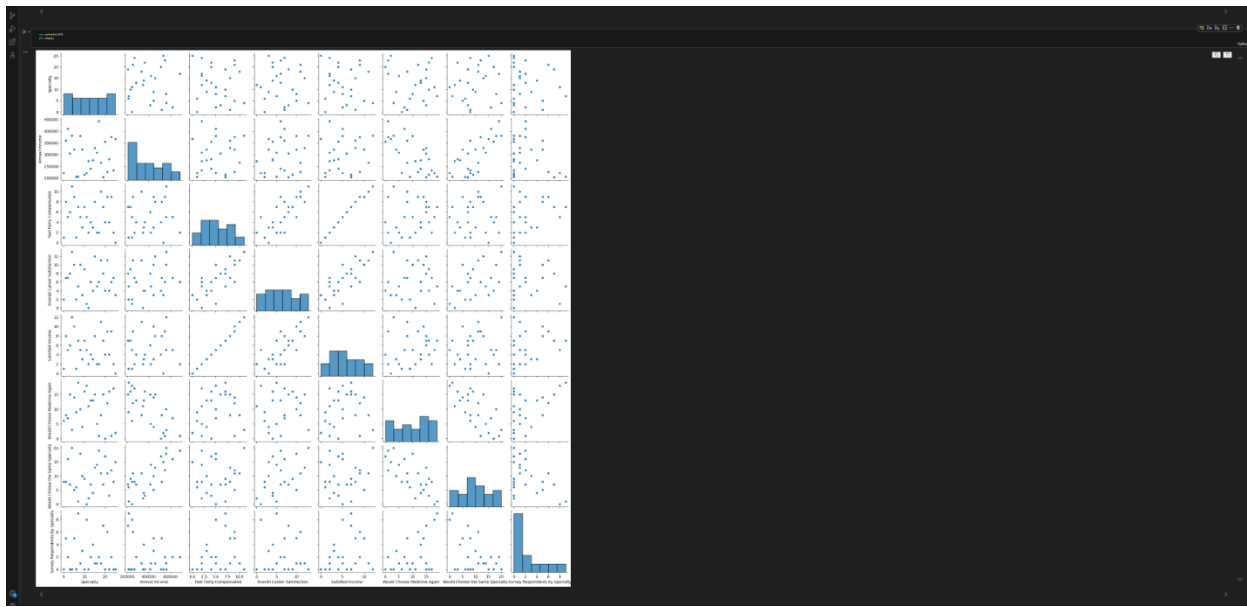


BOXPLOT



PAIRPLOT





```

x = df.drop(['Annual Income'],axis=1)
y = df['Annual Income']

from sklearn.model_selection import train_test_split,GridSearchCV

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state = 42)

from sklearn.linear_model import LinearRegression

reg = LinearRegression()

reg.fit(x_train,y_train)

...
LinearRegression()

from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error

y_train_pred = reg.predict(x_train)
y_test_pred = reg.predict(x_test)

y_train_pred[:5]

...
array([[288992.21158909, 336580.02502935, 381000.
, 263425.25831949,
256673.14018389]])

y_test_pred[:5]

...
array([[279196.68196691, 286854.5193157 , 367767.14729714, 280119.45280465,
326738.00154486]])

r2_score(y_train,y_train_pred) * 100

...
67.9954207297506

```

```
mean_squared_error(y_train,y_train_pred)
... 1485231186.3519106

r2_score(y_test,y_test_pred)*100
... 27.269167796800964

mean_squared_error(y_test,y_test_pred)
... 3715045452.16928

from sklearn.ensemble import RandomForestRegressor

rf=RandomForestRegressor(n_estimators=100,random_state=42)

rf.fit(x_train,y_train)

...
+ RandomForestRegressor
RandomForestRegressor(random_state=42)

y_train_pred=rf.predict(x_train)
y_test_pred=rf.predict(x_test)

r2_score(y_train,y_train_pred)
... 0.8955265319033624

mean_squared_error(y_train,y_train_pred)
... 458713655.5555556

r2_score(y_test,y_test_pred)
```

```
mean_squared_error(y_test,y_test_pred)
... 3631440587.5

from sklearn.tree import DecisionTreeRegressor

dtr=DecisionTreeRegressor(random_state=42)
+ Code + Markdown

dtr.fit(x_train,y_train)

...
+ DecisionTreeRegressor
DecisionTreeRegressor(random_state=42)

y_train_pred=dtr.predict(x_train)
```

```

y_test_pred=dtr.predict(x_test)

y_train_pred[:5]

... array([[207000., 367000., 301000., 306000., 273000.]])

y_test_pred[:5]

... array([[175000., 215000., 355000., 306000., 355000.]])

r2_score(y_train,y_train_pred)*100

... 100.0

mean_squared_error(y_train,y_train_pred)

```

```

mean_squared_error(y_train,y_train_pred)

... 0.0

r2_score(y_test,y_test_pred)*100

... 30.204216476806934

mean_squared_error(y_test,y_test_pred)

... 3565125000.0

import xgboost as xgb

xg_reg=xgb.XGBRegressor()

```

```

xg_reg.fit(x_train,y_train)

...
XGBRegressor(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)

y_train_pred=xg_reg.predict(x_train)
y_test_pred=xg_reg.predict(x_test)

r2_score(y_train,y_train_pred)*100

... 99.99999999996973

mean_squared_error(y_train,y_train_pred)

... 0.0013292100694444445

```

```
mean_squared_error(y_train,y_train_pred)
... 0.0013292100694444445

r2_score(y_test,y_test_pred)*100
... 30.178970363928848

mean_squared_error(y_test,y_test_pred)
... 3566414555.666992

r2_score(y_train,y_train_pred)*100
... 99.999999999996973

reg.predict([[11,5,1,5,10,0,1]])
... array([[229413.33760417]])
```

```
reg.predict([[23,9,6,9,1,12,4]])
... array([[351924.34431502]])

reg.predict([[10,7,9,7,16,9,0]])
... array([[266779.67041667]])
```

