# **INTERNSHIP PROJECT REPORT**

|  |  |
| --- | --- |
| Internship Project Title | HR Salary Dashboard - Train the Dataset and Predict Salary |
| Name of the Company | TCS ION |
| Name of the Industry Mentor | Avinash Singh |
| Name of the Institute | Amity University |

### **Intern Information**

|  |  |
| --- | --- |
| Name of Intern | Arpita Show |
| University Name | Amity University |
| Couse Name | BCA |
| Speciallization | Data Analytical |

# **Acknowledgement**

I would like to express my sincere gratitude to TCS iON for providing me the opportunity to work on this project. I am thankful to my mentors and colleagues for their continuous guidance, feedback, and support throughout the duration of this project. Their expertise and encouragement have been invaluable in the successful completion of this work.

# **Objective**

The primary objective of this project is to build a predictive machine learning model capable of accurately estimating the salary of an employee based on various HR-related features. This will assist the HR department in salary planning, budgeting, and decision-making by leveraging data-driven insights.

# **Introduction**

The HR Salary Prediction project utilizes historical employee data to develop a predictive model for estimating salaries. The dataset contains demographic, departmental, and performance-related information, with 'Salary' being the target variable. Through data cleaning, feature encoding, and model training, the project aims to create a reliable salary prediction tool.

# **Approach / Methodology**

The methodology adopted for this project consists of the following key steps:

1. Data Cleaning and Preprocessing

2. Feature Encoding using LabelEncoder

3. Splitting the dataset into training and testing sets

4. Training the Random Forest Regressor model

5. Model Evaluation using R² Score and MSE

6. Salary Prediction for new employees

7. Feature Importance Analysis

## **Data Cleaning and Preprocessing Code**

import pandas as pd  
 from sklearn.preprocessing import LabelEncoder  
   
df = pd.read\_csv("HRDataset\_v14.csv")  
 columns\_to\_drop = ['Employee\_Name', 'ManagerName', 'DateofTermination',  
 'LastPerformanceReview\_Date', 'TermReason', 'Zip', 'DOB',  
 'DateofHire', 'Age']  
 df.drop(columns=columns\_to\_drop, inplace=True, errors='ignore')  
 df.replace("#VALUE!", pd.NA, inplace=True)  
 df.dropna(inplace=True)  
   
for col in df.select\_dtypes(include='object').columns:  
 df[col] = df[col].str.strip()  
   
label\_encoders = {}  
 for col in df.select\_dtypes(include='object').columns:  
 le = LabelEncoder()  
 df[col] = le.fit\_transform(df[col])  
 label\_encoders[col] = le  
   
emp\_ids = df['EmpID']  
 X = df.drop(['Salary', 'EmpID'], axis=1)  
 y = df['Salary']

## **Model Training and Evaluation**

from sklearn.model\_selection import train\_test\_split  
 from sklearn.ensemble import RandomForestRegressor  
 from sklearn.metrics import r2\_score, mean\_squared\_error  
   
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
   
model = RandomForestRegressor(random\_state=42)  
 model.fit(X\_train, y\_train)  
   
y\_pred = model.predict(X\_test)  
   
r2 = r2\_score(y\_test, y\_pred)  
 mse = mean\_squared\_error(y\_test, y\_pred)  
   
print("R² Score:", r2)  
 print("Mean Squared Error:", mse)

Model trained successfully! R² Score: 0.5033, Mean Squared Error: 562499510.47

## **Salary Prediction for New Employee**

new\_employee\_data = {  
 'MarriedID': [1],  
 'MaritalStatusID': [1],  
 'GenderID': [0],  
 'EmpStatusID': [1],  
 'DeptID': [5],  
 'PerfScoreID': [3],  
 'FromDiversityJobFairID': [0],  
 'Termd': [0],  
 'PositionID': [0],  
 'Position': [20],  
 'State': [3],  
 'Sex': [0],  
 'MaritalDesc': [1],  
 'CitizenDesc': [0],  
 'HispanicLatino': [0],  
 'RaceDesc': [2],  
 'EmploymentStatus': [1],  
 'Department': [0],  
 'ManagerID': [4],  
 'RecruitmentSource': [0],  
 'PerformanceScore': [0],  
 'EngagementSurvey': [2.0],  
 'EmpSatisfaction': [1],  
 'SpecialProjectsCount': [1],  
 'DaysLateLast30': [0],  
 'Absences': [4]  
 }  
 new\_employee\_df = pd.DataFrame(new\_employee\_data)  
 predicted\_salary = model.predict(new\_employee\_df)  
 print("Predicted Salary:", predicted\_salary[0])

Predicted Salary: 77357.07

# **Charts & Algorithms**

Feature Importance Analysis:

import matplotlib.pyplot as plt  
 import seaborn as sns  
   
importances = model.feature\_importances\_  
 feature\_names = X.columns  
   
feat\_imp = pd.DataFrame({'Feature': feature\_names, 'Importance': importances})  
 feat\_imp.sort\_values(by='Importance', ascending=False, inplace=True)  
   
plt.figure(figsize=(10, 6))  
 sns.barplot(x='Importance', y='Feature', data=feat\_imp, palette='viridis')  
 plt.title('Feature Importance for Salary Prediction')  
 plt.xlabel('Importance Score')  
 plt.ylabel('Feature')  
 plt.tight\_layout()  
 plt.show()

# **Challenges & Opportunities**

Challenges faced during the project included handling missing values, encoding categorical features without losing interpretability, and avoiding overfitting of the model. Opportunities lie in refining the model through hyperparameter tuning, incorporating additional relevant features, and expanding the dataset for improved accuracy.

# **Outcome / Conclusion**

The project successfully demonstrated the application of machine learning techniques in HR analytics. The Random Forest Regressor model was able to predict employee salaries with reasonable accuracy (R² score of 0.5033). Feature importance analysis provided valuable insights into the factors influencing salaries.

# **Enhancement Scope**

Future improvements could include implementing hyperparameter optimization, exploring advanced regression models, and integrating the model into a live HR dashboard for real-time salary predictions.

# **Link to Code**

Google Colab Link: <https://colab.research.google.com/drive/1kCU-1rkllCtp8qjkYHemdDXS6HrOXY5c?usp=sharing>

### **Link to Dashboard**

Microsoft 365: [HR Dataset.xlsx](https://1drv.ms/x/c/84fefe525e4ad4f6/ETw0R-WCvptGj-kLPIlE61IBokcKE07FbjWk96JG5rQfqw?e=HYBF8i)