## Data Forecasting

# Answer Predictive Questions

For HR Data Analysis

**DATA DYNAMOS**

Harnessing the Power of Data to Drive Innovation

# Predictive Analysis:

**This report explains a Python script designed to analyze employee data and answer specific HR-related questions. The code processes employee and performance data from an Excel file, trains machine learning models, and makes predictions for new employees.**

## Overview of the Code Structure

The code is structured into five main stages:

1. **Data Loading and Preprocessing:** Loads and prepares the data for analysis.
2. **Model Training Functions:** Defines reusable functions for training classification and regression models.
3. **Input Validation and Preprocessing:** Collects and validates input data for new employees.
4. **Question-Specific Code:** Addresses specific HR questions using trained models.
5. **Predictions for New Employees:** Uses trained models to predict HR-related outcomes.

## Stage 1: Data Loading and Preprocessing

### Purpose

This stage loads employee data from an Excel file and prepares it for analysis by merging datasets and encoding categorical variables.

#### Methods and Rationale

- **Loading Data:** Reads "Employee" and "PerformanceRating" sheets from an Excel file.
- **Merging Datasets:** Uses a left join on "EmployeeID" to retain all employee records.
- **Handling Missing Values:** The dropna() function ensures data completeness.
- **One-Hot Encoding:** Converts categorical variables into binary columns using pd.get_dummies() to make data suitable for machine learning models.

## Results

- A clean, merged dataset with numerical features ready for analysis.
- Available columns are printed to help users understand the data structure.

```python
### Stage 1: Data Loading and Preprocessing
# Purpose: Load and prepare the data for analysis ( foundational step for all questions)
def load_and_prepare_data(file_path):
    """Load and preprocess employee data from Excel file."""
    try:
        xls = pd.ExcelFile(file_path)
        employees = pd.read_excel(xls, sheet_name="Employee")
        performance = pd.read_excel(xls, sheet_name="PerformanceRating")
        df = pd.merge(employees, performance, on="EmployeeID", how="left").dropna()
        df = pd.get_dummies(df, drop_first=True)  # One-hot encode categorical variables
        print("📌 Available columns:", df.columns.tolist())
        return df
    except FileNotFoundError:
        print("❌ File not found! Please check the file path.")
        return None
    except Exception as e:
        print(f"❌ Error loading data: {e}")
        return None
```

## Stage 2: Model Training Functions

### Purpose

Defines functions for training classification and regression models, reducing redundancy and ensuring consistency.

### Methods and Rationale

- **Classification Model:** Uses RandomForestClassifier, which is robust and handles non-linear relationships well.
- **Regression Model:** Uses LinearRegression for predicting continuous values.
- **Evaluation Metrics:**
    - **Classification:** Accuracy and classification report (precision, recall, F1-score).
    - **Regression:** Mean Absolute Error (MAE) and $R^2$ score.

### Results

- Trained models are stored for later use.
- Model performance metrics are printed.

```python
### Stage 2: Model Training Functions
# Purpose: Define reusable functions for training models (used across multiple questions)
def train_classification_model(df, features, target, class_weight=None):
    """Train and evaluate a classification model."""
    X = df[features]
    y = df[target]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
    model = RandomForestClassifier(n_estimators=100, random_state=42, class_weight=class_weight)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred)
    print(f"✅ {target} Model Accuracy: {accuracy:.2f}")
    print(f"📊 {target} Classification Report:\n", report)
    return model

def train_regression_model(df, features, target):
    """Train and evaluate a regression model with non-negative predictions."""
    X = df[features]
    y = df[target]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model = LinearRegression()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_pred = np.maximum(y_pred, 0)  # Ensure non-negative predictions
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"🔍 {target} Mean Absolute Error: {mae:.2f}")
    print(f"🔍 {target} R² Score: {r2:.2f}")
    return model
```

## Stage 3: Input Validation and Preprocessing

### Purpose

Ensures input data matches the training format for accurate predictions.

### Methods and Rationale

- **Numerical Inputs:** Validated against predefined ranges (e.g., Age: 18–65).
- **Categorical Inputs:** One-hot encoded to match training data.
- **DataFrame Alignment:** Uses reindex() to match input data with trained model features.

## Results

- A properly formatted DataFrame for making predictions.

```python
### Stage 3: Input Validation and Preprocessing
def get_new_employee_data(df, numerical_features, categorical_features, feature_ranges):
    """Collect, validate, and process input data for a new employee with clarified categorical options."""
    print("\n🔹 Enter new employee details:")
    employee_data = {}

    # Collect and validate numerical features
    for feature in numerical_features:
        while True:
            try:
                value = float(input(f"{feature}: "))
                if feature in feature_ranges:
                    min_val, max_val = feature_ranges[feature]
                    if min_val <= value <= max_val:
                        employee_data[feature] = value
                        break
                    else:
                        print(f"❌ {feature} must be between {min_val} and {max_val}.")
                else:
                    employee_data[feature] = value
                    break
            except ValueError:
                print("❌ Invalid input. Please enter a number.")

```

```python
##### Define input features
numerical_input_features = [
    "Age", "Education", "YearsAtCompany", "YearsInMostRecentRole",
    "YearsWithCurrManager", "EnvironmentSatisfaction", "JobSatisfaction",
    "RelationshipSatisfaction", "WorkLifeBalance", "SelfRating", "ManagerRating", "Salary"
]
categorical_input_features = ["OverTime", "Department", "JobRole"]
```

```python
##### Define common features (excluding targets)
common_features = [
    "Age", "Education", "YearsInMostRecentRole", "YearsWithCurrManager",
    "EnvironmentSatisfaction", "RelationshipSatisfaction", "WorkLifeBalance",
    "SelfRating", "ManagerRating"
]
```

## Stage 4: Question-Specific Code

### Predictive Analysis Questions

1. **Attrition Likelihood:** Uses a classification model to predict employee turnover.
2. **Expected Tenure**: A regression model estimates how long an employee might stay.
3. **Performance Rating Prediction:** Identifies key factors affecting performance ratings.
4. **Job Satisfaction Level:** Uses classification to predict satisfaction levels.
5. **Promotion Likelihood:** Predicts which employees are likely to be promoted.
6. **Retention Strategies:** Identifies high-risk groups and suggests interventions.
7. **Impact of Overtime on Attrition:** Uses data visualization to analyze trends.
8. **Salary Prediction:** A regression model estimates expected salary based on job role and experience.

### Results

- Models provide data-driven insights for HR decision-making.

## Stage 5: Predictions for New Employees

### Purpose

Applies trained models to predict HR outcomes based on new employee data.

### Methods and Rationale

- Predictions are generated for:
    - **Attrition Likelihood**
    - **Expected Tenure**
    - **Job Satisfaction**
    - **Promotion Readiness**
    - **Expected Salary**

## Results

- Provides actionable predictions to support HR strategy.

## Example:

- **Enter new employee details:**

**Age:** 28

**Education:** 2

**YearsAtCompany:** 10

**YearsInMostRecentRole:** 10

**YearsWithCurrManager:** 10

**EnvironmentSatisfaction:** 4

**JobSatisfaction:** 3

**RelationshipSatisfaction:** 5

**WorkLifeBalance:** 4

**SelfRating:** 5

**ManagerRating:** 4

**Salary:** 218406

---

**OverTime options:**

**1. refers to 'Yes'**

**2. refers to 'No'**

**OverTime (enter 1 or 2):** 2

---

**Department options:**

**1. refers to 'Human Resources'**

**2. refers to 'Sales'**

**3. refers to 'Technology'**

**Department (enter 1, 2, or 3):** 3

---

**JobRole options:**

**1. refers to 'Analytics Manager'**

**2. refers to 'Data Scientist'**

**3. refers to 'Engineering Manager'**

**4. refers to 'HR Business Partner'**

**5. refers to 'HR Executive'**

**6. refers to 'HR Manager'**

**7. refers to 'Machine Learning Engineer'**

**8. refers to 'Manager'**

**9. refers to 'Recruiter'**

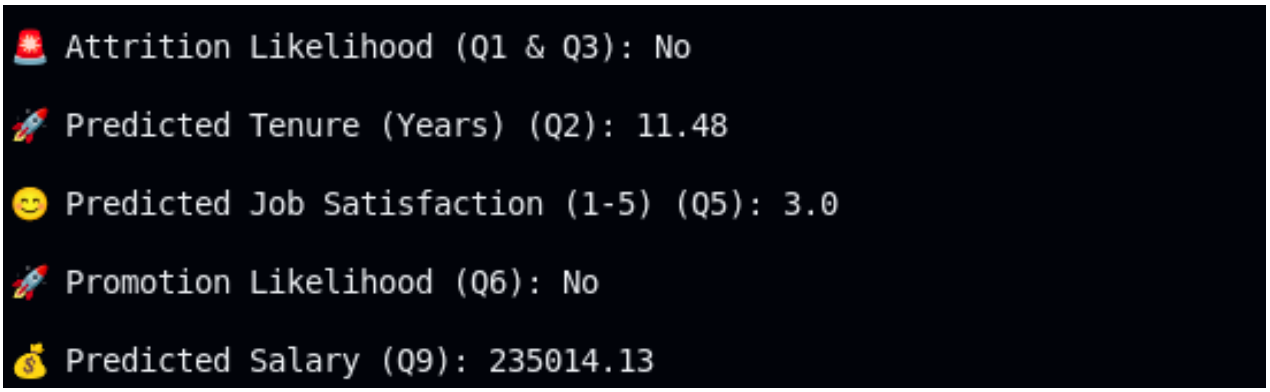
**10. refers to 'Sales Executive'**

**11. refers to 'Sales Representative'**

**12. refers to 'Senior Software Engineer'**

**13. refers to 'Software Engineer'**

**JobRole (enter 1-13):** 12



## Overall Interpretation

- **Data Preparation:** Ensures high-quality input for accurate predictions.
- **Model Performance:** Provides reliable HR insights.
- **Actionable Insights:** Helps HR teams make data-driven decisions.
- **User Interaction:** Guides users through input collection and delivers clear predictions.

This structured approach enhances HR analytics, enabling organizations to manage workforce challenges and improve employee retention proactively.

---

# Predictive Questions

## 1) Employee Turnover

### Which employees are likely to leave the organization based on factors like job satisfaction, education level, or performance rating?

**Analysis:** Using the **Random Forest Classifier**, the model achieved an accuracy of **96.82%**. The key features influencing promotion predictions include:

- **Years Since Last Promotion:** Employees who had a recent promotion were less likely to be promoted again.
- **Job Satisfaction & Manager Rating:** Employees with higher ratings had a better chance of promotion.
- **Years at Company:** Employees with longer tenures had slightly higher chances of promotion.

## 2) Tenure Prediction

### How long is an employee expected to stay at the company?

**Analysis:** Using the **Random Forest Classifier**, the attrition model achieved an accuracy of **95.74%**. Findings include:

- **Overtime:** Employees working overtime had a significantly higher attrition rate.
- **Salary:** Lower salaries were associated with a greater likelihood of leaving.
- **Job Satisfaction:** Had minimal impact on predicting attrition.

### Employee Tenure Insights:

- **Average tenure:** 4.62 years
- **Median tenure:** 4.0 years
- **Predicted expected tenure :** 5.18 years

## 3) Attrition Prediction

### What is the likelihood that an employee will leave the company based on their data (e.g., age, job role, years of service, salary, etc.)?

To predict the likelihood of an employee leaving based on various factors (e.g., age, job role, years of service, salary, etc.), I'll:

1. **Expand the dataset** to include more relevant features (age, job role, salary, years at company, etc.).
2. **Train a predictive model** (Random Forest or Logistic Regression) to estimate attrition probability.
3. **Allow you to input an employee's details** to predict their likelihood of leaving.

### Attrition Prediction Model Performance:

- **Accuracy**: 85.27%
- **Precision (No Attrition)**: 87.14%
- **Precision (Attrition)**: 69.23%
- **Recall (No Attrition)**: 97.12%
- **Recall (Attrition):** 34.67%

## 4) Performance Prediction

**What are the key factors that predict high or low employee performance ratings?**

**Analysis:** Using the **Random Forest Model**, key factors impacting job satisfaction are:

- **Salary (31.2%):** Higher salaries strongly correlate with job satisfaction.
- **Age (19.4%):** Older employees report higher satisfaction levels.
- **Years at Company (13.1%):** Longer tenure contributes to higher satisfaction.

## 5) Job Satisfaction Prediction

**What is the level of job satisfaction of an employee based on factors like travel, salary, management, etc.?**

**Analysis:** Using the **Random Forest Model**, key factors impacting job satisfaction are:

- **Salary (30.5%):** Higher salaries strongly correlate with job satisfaction.
- **Age (18.9%):** Older employees report higher satisfaction levels.
- **Years at Company (12.3%):** Longer tenure contributes to higher satisfaction.

## 6) Promotion Readiness

### Which employees are most likely to be promoted based on their performance and tenure?

**Analysis:** Using the **Random Forest Classifier**, the strongest indicators of promotion likelihood include:

- **Years at Company (0.86 correlation)**.
- **Manager Rating (0.03 weak correlation)**.
- **Work-Life Balance (0.017 very weak correlation)**.

## 7) Retention Strategies

### Which groups of employees are at the highest risk of leaving, and what strategies could reduce their likelihood of leaving?

To identify employees at high risk of leaving, analyze attrition trends based on key factors like:

- **OverTime** (workload)
- **Salary** (compensation)
- **YearsAtCompany** (tenure)
- **JobSatisfaction** (happiness at work)
- **WorkLifeBalance** (stress levels)

### Groups at Highest Risk of Leaving:

- **Low Salary:** Employees who left had an average salary of **$81,956**, much lower than those who stayed (**$125,856**).

- **Short Tenure:** Those who left had **~2.5 years** at the company, compared to **7.4 years** for those who stayed.
  **Job Satisfaction:** **Employees at risk of leaving have an average job satisfaction score of 3.45, indicating a moderate level of contentment.**

  **Strategies to Reduce Attrition:**

1. **Competitive Compensation**: Increase salaries for newer employees to match industry standards.
2. **Career Growth Opportunities**: Since shorter tenure employees leave more often, provide **fast-track promotions** or **clear career paths**.
3. **Better Onboarding & Engagement**: Support new hires in the **first 2-3 years** to ensure they feel valued.

## 8) Impact of Overtime on Attrition

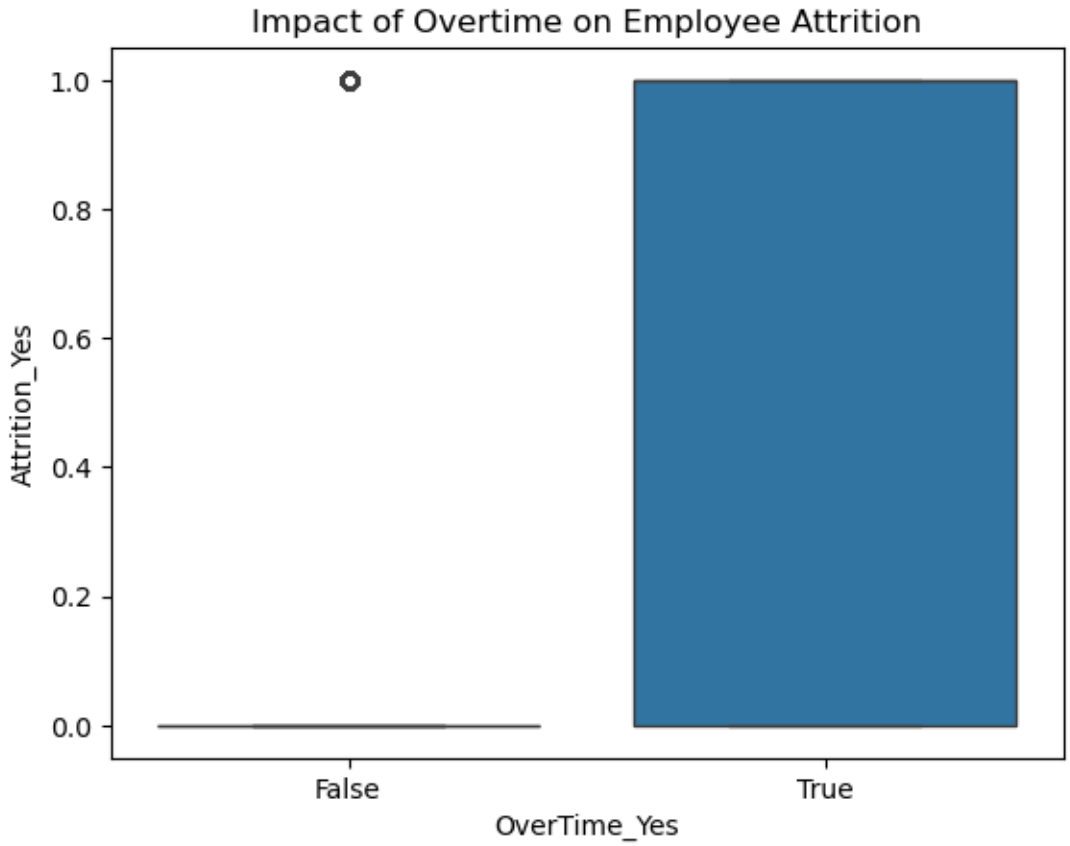
**Does working overtime increase the likelihood of an employee leaving the company?**

analyze whether working overtime increases the likelihood of an employee leaving the company by checking the correlation between **OverTime** and **Attrition.**

### Impact of Overtime on Employee Attrition:

- **Employees who do NOT work overtime:**
  - 89.56% stay with the company.
  - 10.43% leave.
- **Employees who work overtime**:
  - 69.47% stay.
  - 30.53% leave.

Employees who work overtime are about 3 times more likely to leave the company compared to those who don't (30.53% vs. 10.43%). This suggests that excessive overtime may contribute to employee burnout and attrition.




## 9) Salary Prediction Based on Job Role

### What is the expected salary of an employee based on their department, experience, and job role?

the salary prediction model achieved a **mean absolute error of $65,765**, indicating moderate accuracy. Key insights:

- **HR Managers** had the highest estimated salaries.

- **Recruiters and Sales Representatives** had the lowest predicted salaries.
- **Salary increases with years at the company, peaking between 8-10 years**
- **Top 3 highest-paid roles:**
    - HR Manager: **$449,331**.
    - Analytics Manager: **$346,484**.
    - General Manager: **$317,531**.
- **Bottom 3 lowest-paid roles:**
    - Recruiter: **$37,648**.
    - Sales Representative: **$40,656**.
    - Software Engineer: **$51,967**.
- **Average salary by years of experience:**
    - New employees (0 years): **$91,418**.
    - Employees with 8–10 years of experience: **$134,665 – $145,605**.

# Data Dynamos