Digital Nomad Salary Predictor

Estimate your potential salary as a location-independent worker

Job Role

Data Analyst

Location (City, Country)

London, United Kingdom

Productivity (1-10)

7

Burnout Level (1-10)

4

Options

- [x] Remote Company
- [x] Nomad Visa

Predict Salary

Predicted Salary

$94,007.89

Input Features

- → **burnout level:** 4
- → **city:** London
- → **company remote:** 1
- → **country:** United Kingdom
- → **job role:** Data Analyst
- → **location:** London, United Kingdom
- → **nomad visa:** 1
- → **productivity:** 7

templates\index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Digital Nomad Salary Predictor</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        body {
            background-color: #f8f9fa;
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        }
        .container {
            max-width: 800px;
            margin-top: 50px;
        }
        .card {
            border-radius: 15px;
            box-shadow: 0 6px 20px rgba(0, 0, 0, 0.1);
            border: none;
        }
        .card-header {
            background-color: #6c63ff;
            color: white;
            border-radius: 15px 15px 0 0 !important;
            padding: 20px;
            text-align: center;
        }
        .btn-predict {
            background-color: #6c63ff;
            border: none;
            padding: 10px 25px;
            font-weight: 600;
        }
        .btn-predict:hover {
            background-color: #5a52d6;
        }
        .result-container {
            display: none;
            margin-top: 30px;
            animation: fadeIn 0.5s;
        }
        @keyframes fadeIn {
            from { opacity: 0; }
            to { opacity: 1; }
        }
        .feature-icon {
            color: #6c63ff;
            margin-right: 10px;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="card">
            <div class="card-header">
                <h2>Digital Nomad Salary Predictor</h2>
```

```
                <p class="mb-0">Estimate your potential salary as a location-independent worker</p>
            </div>
            <div class="card-body">
                <form id="salaryForm">
                    <div class="row mb-3">
                        <div class="col-md-6">
                            <label for="job_role" class="form-label">Job Role</label>
                            <input type="text" class="form-control" id="job_role" required
placeholder="e.g., Data Scientist, Web Developer">
                        </div>
                        <div class="col-md-6">
                            <label for="location" class="form-label">Location (City, Country)</label>
                            <input type="text" class="form-control" id="location" required
placeholder="e.g., Lisbon, Portugal">
                        </div>
                    </div>

                    <div class="row mb-3">
                        <div class="col-md-4">
                            <label for="productivity" class="form-label">Productivity (1-10)</label>
                            <input type="number" class="form-control" id="productivity" min="1"
max="10" required>
                        </div>
                        <div class="col-md-4">
                            <label for="burnout_level" class="form-label">Burnout Level (1-10)
</label>
                            <input type="number" class="form-control" id="burnout_level" min="1"
max="10" required>
                        </div>
                        <div class="col-md-4">
                            <label class="form-label">Options</label>
                            <div class="d-flex">
                                <div class="form-check me-3">
                                    <input class="form-check-input" type="checkbox"
id="company_remote">
                                    <label class="form-check-label" for="company_remote">Remote
Company</label>
                                </div>
                                <div class="form-check">
                                    <input class="form-check-input" type="checkbox" id="nomad_visa">
                                    <label class="form-check-label" for="nomad_visa">Nomad
Visa</label>
                                </div>
                            </div>
                        </div>
                    </div>

                    <div class="text-center">
                        <button type="submit" class="btn btn-predict btn-lg text-white">Predict
Salary</button>
                    </div>
                </form>

                <div id="resultContainer" class="result-container text-center">
                    <h4 class="mb-4">Predicted Salary</h4>
                    <div class="display-4 text-primary mb-4" id="predictedSalary">$0</div>

                    <div class="card mt-4">
                        <div class="card-header bg-light">
                            <h5 class="mb-0">Input Features</h5>
                        </div>
```

```
                    <div class="card-body text-start">
                        <ul class="list-group list-group-flush" id="featureList">
                            <!-- Features will be added here by JavaScript -->
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
</script>
    <script>
        document.getElementById('salaryForm').addEventListener('submit', function(e) {
            e.preventDefault();

            // Get form values
            const job_role = document.getElementById('job_role').value;
            const location = document.getElementById('location').value;
            const productivity = document.getElementById('productivity').value;
            const burnout_level = document.getElementById('burnout_level').value;
            const company_remote = document.getElementById('company_remote').checked ? 'Y' : 'N';
            const nomad_visa = document.getElementById('nomad_visa').checked ? 'Y' : 'N';

            // Create payload
            const payload = {
                job_role,
                company_remote,
                location,
                nomad_visa,
                productivity,
                burnout_level
            };

            // Send to backend
            fetch('/predict', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify(payload)
            })
            .then(response => response.json())
            .then(data => {
                if (data.error) {
                    alert('Error: ' + data.error);
                    return;
                }

                // Display results
                document.getElementById('predictedSalary').textContent =
                    '$' + data.predicted_salary_usd.toLocaleString();

                // Display features
                const featureList = document.getElementById('featureList');
                featureList.innerHTML = '';

                for (const [key, value] of Object.entries(data.input_features)) {
                    const li = document.createElement('li');
```

```
                    li.className = 'list-group-item d-flex align-items-center';
                    li.innerHTML = `
                        <span class="feature-icon">→</span>
                        <strong>${key.replace('_', ' ')}:</strong>  ${value}
                    `;
                    featureList.appendChild(li);
                }

                // Show result container
                document.getElementById('resultContainer').style.display = 'block';
            })
            .catch(error => {
                console.error('Error:', error);
                alert('An error occurred. Please try again.');
            });
        });
    </script>
</body>
</html>
```

train_model.py

```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
import joblib
import warnings
warnings.filterwarnings('ignore')

# Load dataset (replace with your CSV path)
df = pd.read_csv('digital_nomad_salaries.csv')

# Data Cleaning
df = df.drop(['user_id', 'timestamp'], axis=1)  # Remove non-predictive columns
df['company_remote'] = df['company_remote'].map({'Y': 1, 'N': 0})
df['nomad_visa'] = df['nomad_visa'].map({'Y': 1, 'N': 0})

# Feature Engineering
df['city'] = df['location'].apply(lambda x: x.split(',')[0].strip())
df['country'] = df['location'].apply(lambda x: x.split(',')[1].strip())

# Preprocessing pipeline
categorical_features = ['job_role', 'city', 'country']
numeric_features = ['productivity', 'burnout_level', 'company_remote', 'nomad_visa']

preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features),
        ('num', 'passthrough', numeric_features)
    ])

# Model Pipeline
model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(n_estimators=100, random_state=42))
])

# Train-Test Split
X = df.drop('salary_usd', axis=1)
y = df['salary_usd']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model.fit(X_train, y_train)

# Save model and preprocessing pipeline
joblib.dump(model, 'nomad_salary_model.joblib')

print(f"Model trained and saved. R2 Score: {model.score(X_test, y_test):.2f}")

```

app.py

```python
from flask import Flask, request, jsonify, render_template
import joblib
import pandas as pd
import numpy as np

app = Flask(__name__)

# Load model
model = joblib.load('nomad_salary_model.joblib')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict_salary():
    try:
        # Get JSON data
        data = request.json

        # Validate required fields
        required_fields = [
            'job_role', 'company_remote', 'location',
            'nomad_visa', 'productivity', 'burnout_level'
        ]

        if not all(field in data for field in required_fields):
            return jsonify({"error": "Missing required fields"}), 400

        # Prepare input data
        input_data = {
            'job_role': data['job_role'],
            'company_remote': 1 if str(data['company_remote']).upper() == 'Y' else 0,
            'location': data['location'],
            'nomad_visa': 1 if str(data['nomad_visa']).upper() == 'Y' else 0,
            'productivity': int(data['productivity']),
            'burnout_level': int(data['burnout_level']),
            'city': data['location'].split(',')[0].strip(),
            'country': data['location'].split(',')[1].strip()
        }

        # Convert to DataFrame
        input_df = pd.DataFrame([input_data])

        # Make prediction
        prediction = model.predict(input_df)

        return jsonify({
            'predicted_salary_usd': round(float(prediction[0]), 2),
            'input_features': input_data
        })

    except Exception as e:
        return jsonify({"error": str(e)}), 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

TERMINAL CODE :

```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

D:\College 6th Sem\DMA\Practicals\Prac_14>pip install -r
requirements.txt
Collecting flask==2.3.2 (from -r requirements.txt (line 1))
  Downloading Flask-2.3.2-py3-none-any.whl.metadata (3.7 kB)
Collecting scikit-learn==1.2.2 (from -r requirements.txt (line 2))
  Downloading scikit_learn-1.2.2-cp311-cp311-win_amd64.whl.metadata
(11 kB)
Collecting pandas==2.0.3 (from -r requirements.txt (line 3))
  Downloading pandas-2.0.3-cp311-cp311-win_amd64.whl.metadata (18
kB)
Collecting numpy==1.24.3 (from -r requirements.txt (line 4))
  Downloading numpy-1.24.3-cp311-cp311-win_amd64.whl.metadata (5.6
kB)
Collecting joblib==1.2.0 (from -r requirements.txt (line 5))
  Downloading joblib-1.2.0-py3-none-any.whl.metadata (5.3 kB)
Collecting gunicorn==20.1.0 (from -r requirements.txt (line 6))
  Downloading gunicorn-20.1.0-py3-none-any.whl.metadata (3.8 kB)
Collecting Werkzeug>=2.3.3 (from flask==2.3.2->-r requirements.txt
(line 1))
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Requirement already satisfied: Jinja2>=3.1.2 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from flask==2.3.2->-r requirements.txt (line 1)) (3.1.6)
Collecting itsdangerous>=2.1.2 (from flask==2.3.2->-r
requirements.txt (line 1))
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
```

```
Collecting click>=8.1.3 (from flask==2.3.2->-r requirements.txt
(line 1))
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting blinker>=1.6.2 (from flask==2.3.2->-r requirements.txt
(line 1))
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Requirement already satisfied: scipy>=1.3.2 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from scikit-learn==1.2.2->-r requirements.txt (line 2))
(1.14.1)
Collecting threadpoolctl>=2.0.0 (from scikit-learn==1.2.2->-r
requirements.txt (line 2))
  Downloading threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from pandas==2.0.3->-r requirements.txt (line 3))
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from pandas==2.0.3->-r requirements.txt (line 3)) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from pandas==2.0.3->-r requirements.txt (line 3)) (2023.3)
Requirement already satisfied: setuptools>=3.0 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from gunicorn==20.1.0->-r requirements.txt (line 6))
(76.0.0)
Requirement already satisfied: colorama in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from click>=8.1.3->flask==2.3.2->-r requirements.txt (line
1)) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from Jinja2>=3.1.2->flask==2.3.2->-r requirements.txt
(line 1)) (3.0.2)
```

```
Requirement already satisfied: six>=1.5 in
c:\users\divya\appdata\local\programs\python\python311\lib\site-
packages (from python-dateutil>=2.8.2->pandas==2.0.3->-r
requirements.txt (line 3)) (1.17.0)

Downloading Flask-2.3.2-py3-none-any.whl (96 kB)

Downloading scikit_learn-1.2.2-cp311-cp311-win_amd64.whl (8.3 MB)

   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.3/8.3 MB 1.5 MB/s eta
0:00:00

Downloading pandas-2.0.3-cp311-cp311-win_amd64.whl (10.6 MB)

   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.6/10.6 MB 1.6 MB/s
eta 0:00:00

Downloading numpy-1.24.3-cp311-cp311-win_amd64.whl (14.8 MB)

   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.8/14.8 MB 2.2 MB/s
eta 0:00:00

Downloading joblib-1.2.0-py3-none-any.whl (297 kB)

Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)

Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)

Downloading click-8.1.8-py3-none-any.whl (98 kB)

Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)

Downloading threadpoolctl-3.6.0-py3-none-any.whl (18 kB)

Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)

Installing collected packages: Werkzeug, threadpoolctl, numpy,
joblib, itsdangerous, gunicorn, click, blinker, pandas, flask,
scikit-learn

  Attempting uninstall: numpy

    Found existing installation: numpy 1.25.0

    Uninstalling numpy-1.25.0:

      Successfully uninstalled numpy-1.25.0

  Attempting uninstall: pandas

    Found existing installation: pandas 2.2.3

    Uninstalling pandas-2.2.3:
```

```
      Successfully uninstalled pandas-2.2.3
Successfully installed Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8
flask-2.3.2 gunicorn-20.1.0 itsdangerous-2.2.0 joblib-1.2.0 numpy-
1.24.3 pandas-2.0.3 scikit-learn-1.2.2 threadpoolctl-3.6.0


D:\College 6th Sem\DMA\Practicals\Prac_14>python train_model.py
Model trained and saved. R2 Score: 0.09


D:\College 6th Sem\DMA\Practicals\Prac_14>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.220.13:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 115-406-314
192.168.220.13 - - [22/Apr/2025 02:02:17] "GET / HTTP/1.1" 200 -
192.168.220.13 - - [22/Apr/2025 02:02:18] "GET /favicon.ico
HTTP/1.1" 404 -
192.168.220.13 - - [22/Apr/2025 02:03:19] "POST /predict HTTP/1.1"
200 -
192.168.220.13 - - [22/Apr/2025 02:04:16] "GET / HTTP/1.1" 200 –
```