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}")

```