Health Insurance Premium Prediction using Random Forest

Title

Health Insurance Premium Prediction using Random Forest

Abstract / Objective

This project builds a machine learning model to estimate individual health insurance premiums from demographic and lifestyle features. Using a Random Forest regression model trained on real-world insurance data, the objective is to provide accurate premium estimates, surface the strongest cost drivers (e.g., smoking, age, BMI), and produce interpretable outputs that insurers can use to support fair pricing and explainability.

Problem Statement

Health insurance charges vary widely across individuals due to multiple interacting risk factors. Insurers need reliable, transparent models to price policies fairly, while customers and regulators seek interpretable explanations for premiums. The problem addressed here is: **Given individual-level demographic and health features, predict the insurance charge (premium)** with high accuracy and interpretability.

Dataset & Tools Used

Dataset

- Source referenced in notebook: `Health_insurance.csv` (notebook path: `C:/Users/Harsh/Downloads/Health_insurance.csv`).
- This is the commonly used *Medical Cost Personal Dataset* (public Kaggle dataset). Typical dataset size: **1,338 rows × 7 columns** (age, sex, bmi, children, smoker, region, charges) — the notebook uses a subset of columns for modeling.
- Target variable: `charges` (insurance premium).

Tools & Libraries

- Python (Jupyter Notebook)
- pandas, numpy
- plotly.express (for interactive EDA visuals)
- scikit-learn (train_test_split, RandomForestRegressor, metrics)

- (Optional visualization: Matplotlib / Plotly exports for reporting; Power BI screenshots suggested for dashboards)

Data Cleaning & Feature Engineering

(steps performed in the notebook, summarized)

1. **Data loading**
   - `pd.read_csv("C:/Users/Harsh/Downloads/Health_insurance.csv")`
2. **Initial inspection**
   - `data.head()` and visual EDA using `plotly.express` (histograms and pie charts).
   - The notebook checks distributions (sex, smoker, region) and shows plots for visual inspection.
3. **Encoding**
   - `sex` and `smoker` converted to numeric:
   - `data["sex"] = data["sex"].map({"female": 0, "male": 1})`
   - `data["smoker"] = data["smoker"].map({"no": 0, "yes": 1})`
   - No explicit missing-value imputation reported (the notebook indicates dataset ready for modeling).
4. **Feature selection**
   - The model was trained using a **reduced feature set** (explicit in the notebook):
   - `x = np.array(data[["age", "sex", "bmi", "smoker"]])`
   - `y = np.array(data["charges"])`

     So this implementation **used 4 features**: `age`, `sex` (encoded), `bmi`, and `smoker` (encoded).

   - Note: `children` and `region` exist in dataset but were not used in the final model cell shown.
5. **Train/Test split**
   - 80/20 split with fixed seed:
   - `from sklearn.model_selection import train_test_split`
   - `xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)`

Methodology / Model Building

1. **Model choice**
   - A `RandomForestRegressor` (scikit-learn) — chosen in the notebook for robustness and handling nonlinearity.
   - The notebook instantiates with default parameters:
   - `from sklearn.ensemble import RandomForestRegressor`
   - `forest = RandomForestRegressor()`
   - `forest.fit(xtrain, ytrain)`
2. **Prediction**
   - Predictions on test set:
   - `ypred = forest.predict(xtest)`

- Sample predicted values (from notebook output):
- `Predicted Premium Amount`
- `0   10619.250366`
- `1    5383.948626`
- `2   28374.053399`
- `3    9758.047726`
- `4   34571.613695`

3. **Evaluation metrics**
   - Computed in notebook:
   - `from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score`
   - `mae = mean_absolute_error(ytest, ypred)`
   - `mse = mean_squared_error(ytest, ypred)`
   - `r2 = r2_score(ytest, ypred)`
   - `print(f'MAE: {mae}')`
   - `print(f'MSE: {mse}')`
   - `print(f'R2 Score: {r2}')`
   - **Results (as produced in the notebook):**
     - **MAE:** 2658.1378584359704
     - **MSE:** 24,032,832.12725537
     - **R^2:** 0.8451978840900942

   These values indicate the model explains ~84.5% of the variance on the test set and has an average absolute prediction error of ~₹2,658 (units are dataset currency).

Results & Evaluation Metrics (Interpretation)

- **$R^2 \approx 0.845$** — strong explanatory power for the chosen feature set (age, sex, BMI, smoker).
- **MAE ≈ 2,658** — on average, predicted charges differ from actual charges by this magnitude; acceptable depending on business tolerance (for high variability charges, MAE should be contextualized against average charge).
- **MSE is large** due to the squared term emphasizing large prediction errors — expected with heavy-tail distributions in charges.

Key model behaviors observed

- Predictions vary widely (some predicted charges > ₹34k) — compatible with dataset having high-cost outliers (smokers, high BMI, older ages).
- Using only four features still yields strong R^2; this suggests `smoker`, `age`, and `bmi` capture most of the signal in this dataset.

Key Insights & Business Impact

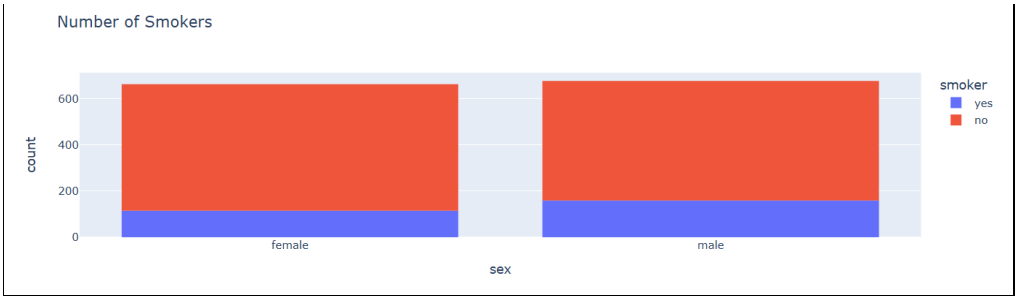
1. **Smoking status is a dominant predictor** — smokers consistently have higher predicted premiums based on EDA and model results. This aligns with domain expectations (smoking => higher health risk => higher claims => higher premium).

2. **Age and BMI show positive relationships** with premium amount — older individuals and higher BMI correlate with higher charges.
3. **Model utility**
    - Insurance underwriters can use the model for quick screening and pricing guidance.
    - Integrating model outputs into dashboards (Power BI or a web UI) can enable actionable policy pricing decisions and scenario analysis (e.g., "what-if" changing BMI or smoking status).
    - Customers can receive transparent explanations for pricing and identify main contributors to their premiums.
4. **Simplicity vs. performance**
    - A compact feature set achieved high R^2 — benefits: faster inference, simpler auditing and explainability.
    - Trade-off: excluding variables like `children` and `region` could miss small but actionable signals for segmented pricing.

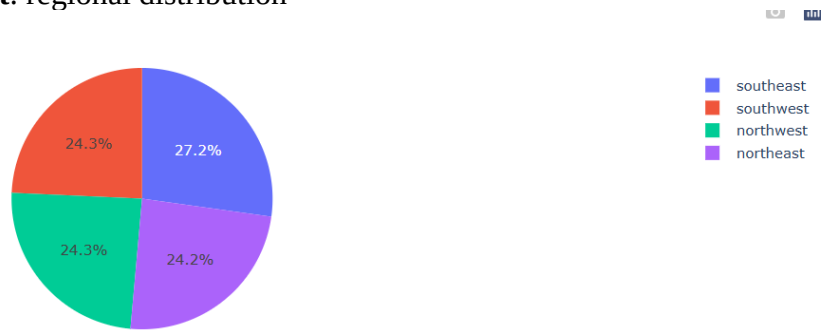
Visualizations (where to add)

1. **Histogram**: count of smokers vs non-smokers by sex —

   "Histogram: smokers vs non-smokers (Plotly)" :-



2. **Pie chart**: regional distribution



Limitations & Future Work

Limitations (noted from the notebook)

- **Limited input features**: the notebook used only `age`, `sex`, `bmi`, and `smoker`. Important variables such as `children`, `region`, socio-economic indicators, or medical history may improve predictions.
- **No hyperparameter search reported**: the Random Forest used the default constructor in the notebook; hyperparameter tuning (GridSearchCV / RandomizedSearchCV) could boost performance and reduce MAE.

References

- scikit-learn `RandomForestRegressor` documentation
- plotly.express documentation

Appendix — Exact Notebook Implementation Notes (from your `.ipynb`)

- Data loaded from: `"C:/Users/Harsh/Downloads/Health_insurance.csv"`.
- Encoding used:
- `data["sex"] = data["sex"].map({"female": 0, "male": 1})`
- `data["smoker"] = data["smoker"].map({"no": 0, "yes": 1})`
- Features used for modeling: `["age", "sex", "bmi", "smoker"]`.
- Train/test split: `test_size=0.2`, `random_state=42`.
- Model: `RandomForestRegressor()` (default scikit-learn parameters).
- Predictions printed in the notebook (sample values shown).
- Metrics reported in notebook:
  - `MAE: 2658.1378584359704`
  - `MSE: 24032832.12725537`
  - `R2 Score: 0.8451978840900942`
