Technical Report: Insurance Charges Prediction

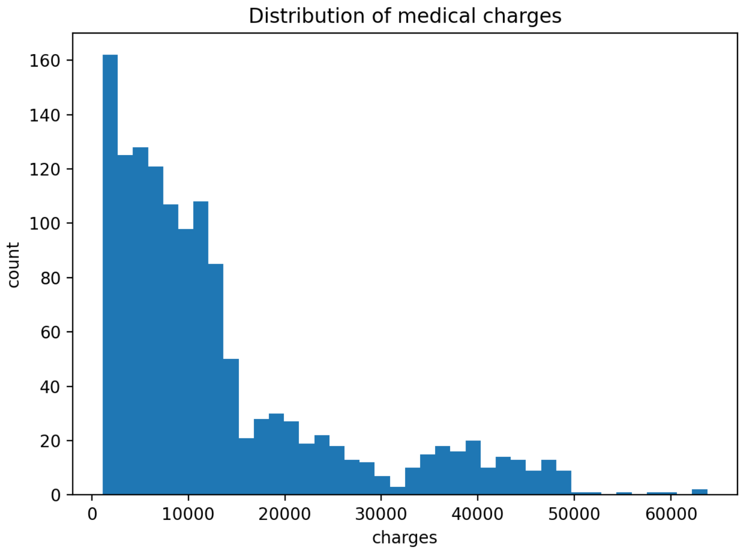
Goal: build an end-to-end Machine Learning pipeline that predicts medical insurance charges from demographic and lifestyle factors, then deploy the best model as a FastAPI service and a simple Streamlit UI.

Dataset

Rows: 1338 | Columns: 7 | Target: charges (continuous). Features: age, sex, bmi, children, smoker, region.

Quick EDA findings

- No missing values in this dataset (0 missing across all columns).
- Charges are right-skewed: min=1121.87, median=9382.03, max=63770.43.
- Smoker status is a strong driver of charges (smokers have much higher median charges).



Pipeline, Models, and Evaluation

Data Validation (Schema)

We enforce a schema: required columns, allowed categorical values, and valid ranges for numeric fields. The same validation is reused in training and in the API to protect the model from bad inputs.

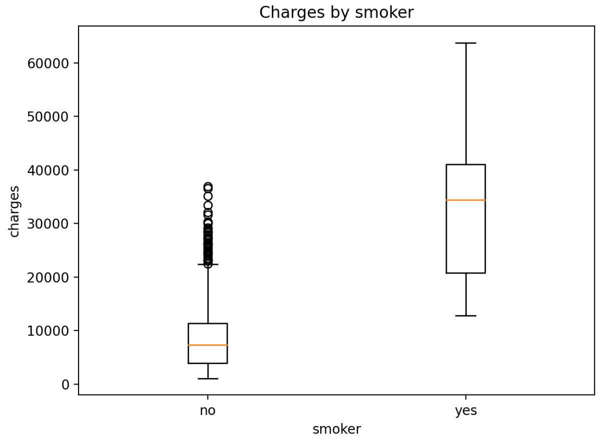
Preprocessing & Feature Engineering

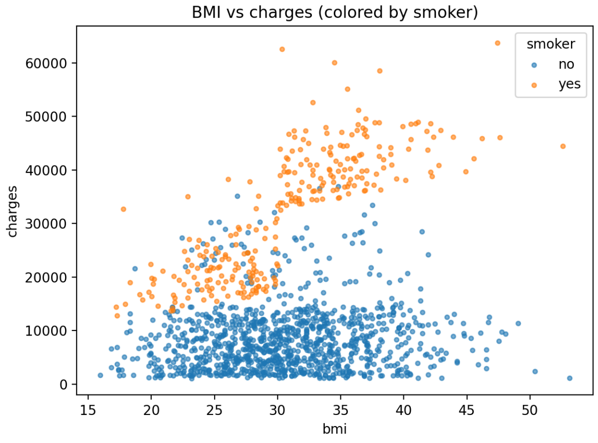
Categorical variables are One-Hot Encoded. Numeric variables are imputed (median) and scaled with StandardScaler. Feature engineering adds BMI category, age group, a smoker indicator, and interaction terms (BMI×smoker, age×smoker).

Model comparison on test set

Model	MAE	MSE	R2
random_forest	2430.35	19754043.29	0.873
ridge	2767.23	20624971.86	0.867
linear_regression	2762.71	20700442.41	0.867

Selected model for deployment: random_forest.





Serving & Product Demo

FastAPI

The service exposes GET /health and POST /predict (JSON). It loads the saved sklearn Pipeline (feature engineering + preprocessing + model) and returns predicted charges.

Streamlit UI

The Streamlit app collects user inputs and calls the API endpoint. This demonstrates a complete workflow from user interface to backend model inference.

How to run

- pip install -r requirements.txt
- python -m src.train
- uvicorn app.api:app --reload
- streamlit run app/web_app.py

Future improvements

- Add cross-validation and hyperparameter tuning (GridSearchCV).
- Experiment with additional models (Gradient Boosting) or a small neural network (bonus).
- Add logging/monitoring and a more robust validation library (e.g., pandera) if allowed.