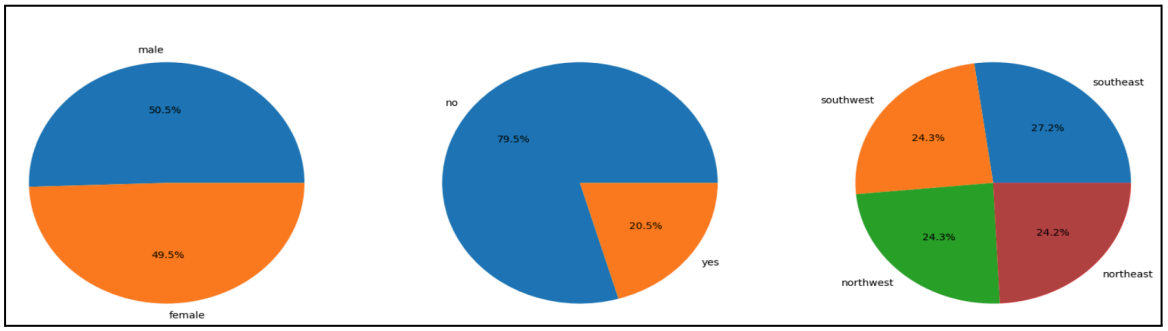
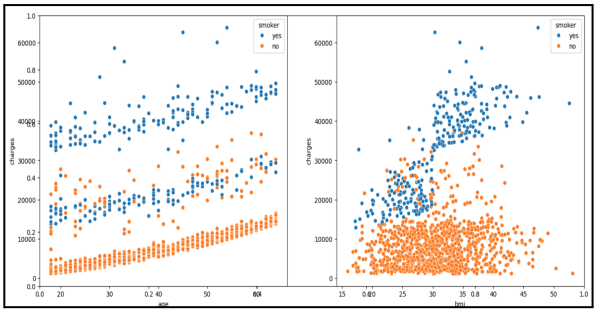
Medical Insurance Price Prediction

Ashish Raj (220101020) Aryan Raj (220101019) Aditya Nanda (220101006)

We try to analyze the dataset which has the premium charges of insurance of its customers having different demographic distributions. The features are age, sex, bmi, children, smoker and region. There are no null or missing values in the dataset. First we plot various graphs and charts to get the insight. Then, We generate the correlation matrix for the dataset. The data is equally distributed among the sex and the region columns but in the smoker column, we can observe a ratio of 80:20. We find that the features age, bmi and smoker are the most influencing feature for the charges of insurance.



	age	sex	bmi	children	smoker	region	charges
age	1.000000	0.020856	0.109272	0.042469	-0.025019	0.003243	0.299008
sex	0.020856	1.000000	-0.046371	-0.017163	-0.076185	-0.007974	-0.057292
bmi	0.109272	-0.046371	1.000000	0.012759	0.003750	0.156686	0.198341
children	0.042469	-0.017163	0.012759	1.000000	0.007673	-0.001907	0.067998
smoker	-0.025019	-0.076185	0.003750	0.007673	1.000000	0.013246	0.787251
region	0.003243	-0.007974	0.156686	-0.001907	0.013246	1.000000	0.011741
charges	0.299008	-0.057292	0.198341	0.067998	0.787251	0.011741	1.000000



We then applied necessary scaling, outlier handling, and encoding of categorical features.

Model 1(Linear Regression with polynomial features and L2 regularization): We created custom feature columns along with the original ones in two degree to capture the non linearity of the data. Then we tried with 3 degree polynomial features. The results were almost the same. The model gave the test score of **0.81 R2 score**. However, We needed to train to large number of epochs and the model could not be optimized further and gave a constant max score of **0.81**.
The model fails to capture the categorical features as well as the multi-dimensionality and the constant score after large number of epochs suggest possible overfitting.

Model 2(Random Forest regressor): Random Forest is an ensemble bagging technique that interprets the data by creating multiple decision trees and averaging their predictions. We use the decision tree algorithm which recursively builds tree based on best split formula which try to get the optimum split by taking the maximum variance difference between parent and (left+right) children. The algorithm also calculates the feature importance. As, The decision tree algorithm is prone to high variance and overfitting, while training our Random forest we take the cross validation score into account which tests the given decision tree on data out of its sample set. We also apply grid cv search algorithm with a set of hyperparameters to find the best set of hyperparameter amongst them.

The random forest regressor gives a test **R2 score of 0.88** and **cross validation score of 0.84.**
- The performance of the model suggests that it captures the overall complexity of data very well. Later we will discuss that it outperforms the XGboost score slightly.

Model 3(XGBoost regressor): XGBoost is also an ensemble technique but works on boosting technique. We first build a random decision tree then try to turn this weak predictor into a strong predictor by penalising for the error in the previous model. We apply the gradient descent on the error to improve our model. The model applies regularisation and takes the weights of data into account. It helps to generalise the whole dataset while avoiding possible overfitting.
It gives a test **R2 score of 0.878** which is slightly lower than the Random Forest regressor. In an ideal situation It should perform better. There are few reasons for this. Model 2 does not account for any regularization thus may capture the variance more accurately. Decreasing the regularization parameter may improve the score for model 3. Also, the model has a lot of hyperparameters which needs to be tuned to get the best performance. Our implemented version is less complex than the library function which overall reduces the performance of the model. **Possible improvements** can be : Since our XGBoost implementation is built from scratch for the project, we did not utilize **Dmatrix,** which is a highly optimized internal data structure used in the official XGBoost library. Re-implementing our model using **Dmatrix** (via the official API) could significantly improve computational efficiency, especially for large-scale or sparse datasets, due to its efficient memory layout, built-in parallelization, and support for missing values.

Comparison with LLM prompting: First we gave the LLM the whole csv dataset and prompted to analyze the data and give predictions on some test set. The LLM interpreted it and gave a generalized mathematical function based prediction which gave an **R2 score of -0.19**. **Then,** We prompted it to apply advanced techniques and models and give a more robust prediction. The result is as follows:

Model	RMSE (Lower is Better)	R^2 Score (Higher is Better)
Linear Regression	5799.59	0.7833
Random Forest	4590.57	0.8643
Gradient Boosting	4352.54	0.8780

These are the train score for the whole dataset. As, We can observe that our trained models outperforms the LLM results for all the three models.
Notebook Links:
Model 1: Copy of 11Yet another copy of MlProject.ipynb
Model 2: Copy of Untitled0.ipynb
Model 3: Another copy of Medical Insurance Price Prediction using Machine Learning –…