Health Insurance Prediction

Aitzaz Tahir Ch
19p-0012

—

Sir Musadaq Mansoor

—

Data Science

Overview

In this project we are solving the regression problem in which it consists of calculating the health insurance charge in the United States (which is our data set).

We are using XGBOOST here. It is one of the most powerful algorithms within Machine Learning, it generates interesting results in such a short time. It is the winner of multiple competitions on the kaggle platform.

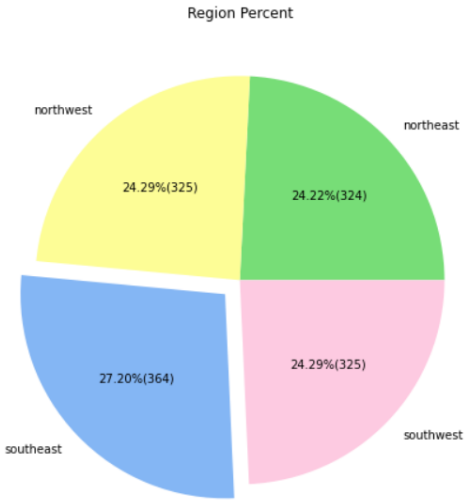
THE PROCESS

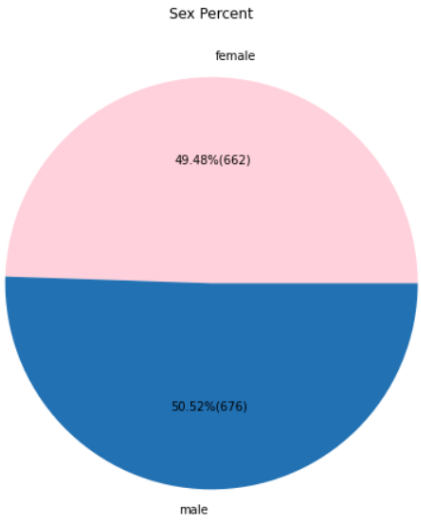
Features of dataset

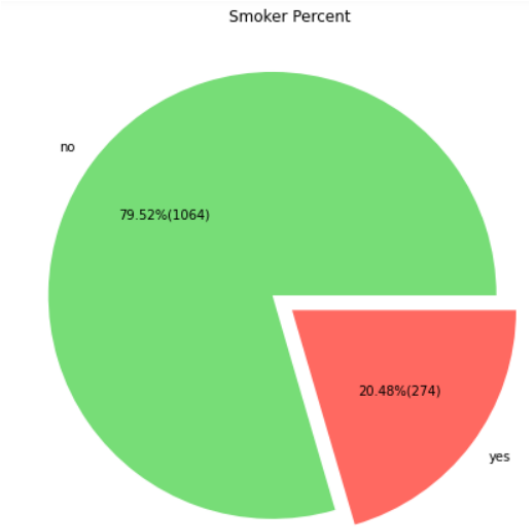
- Age of the insured.
- BMI body mass index.
- Children Number of children of the insured.
- Region User's place of residence.
- Smoker Whether the user smokes or not.
- Charges Health insurance price.

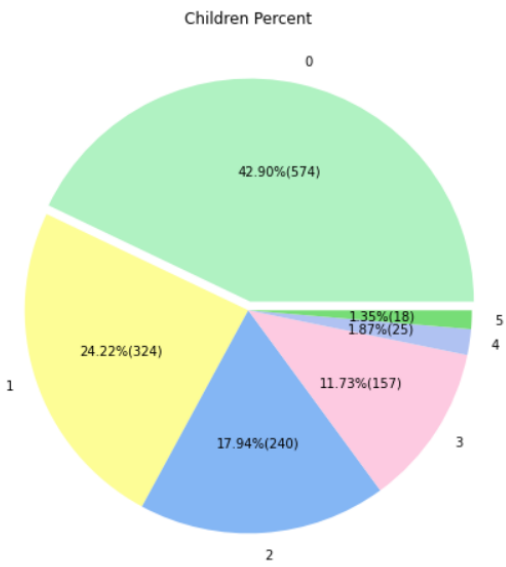
	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

Data Visualization

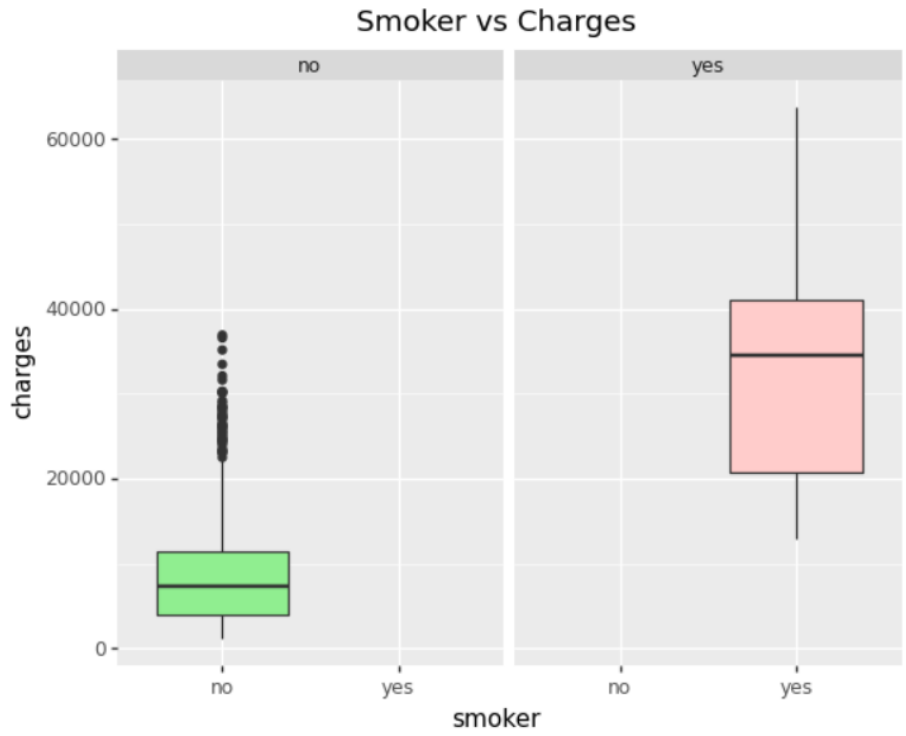






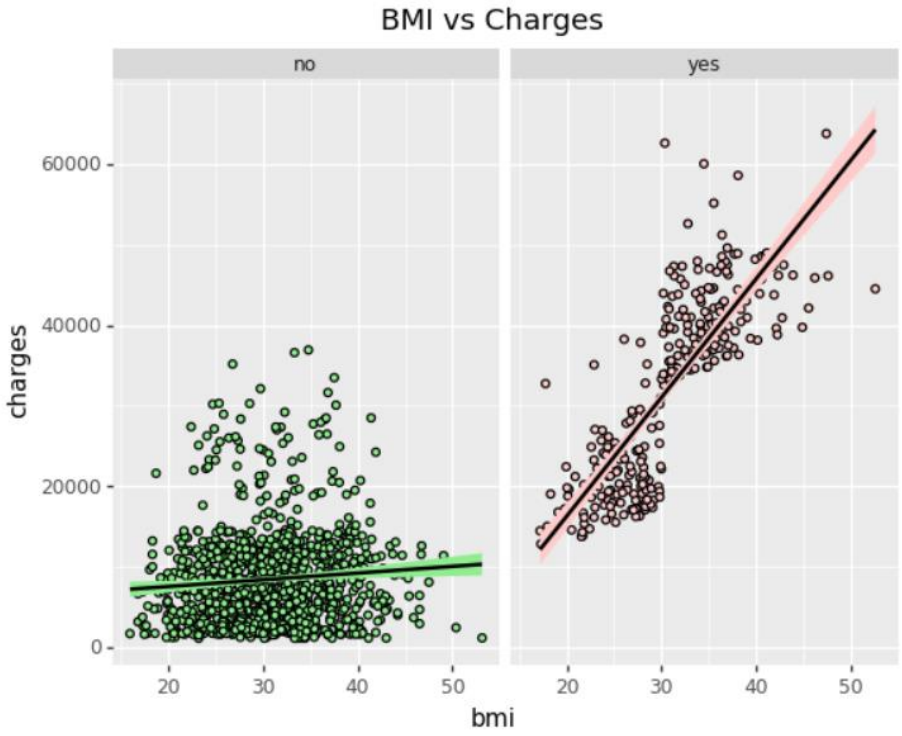


The price of insurance is higher for people who smoke

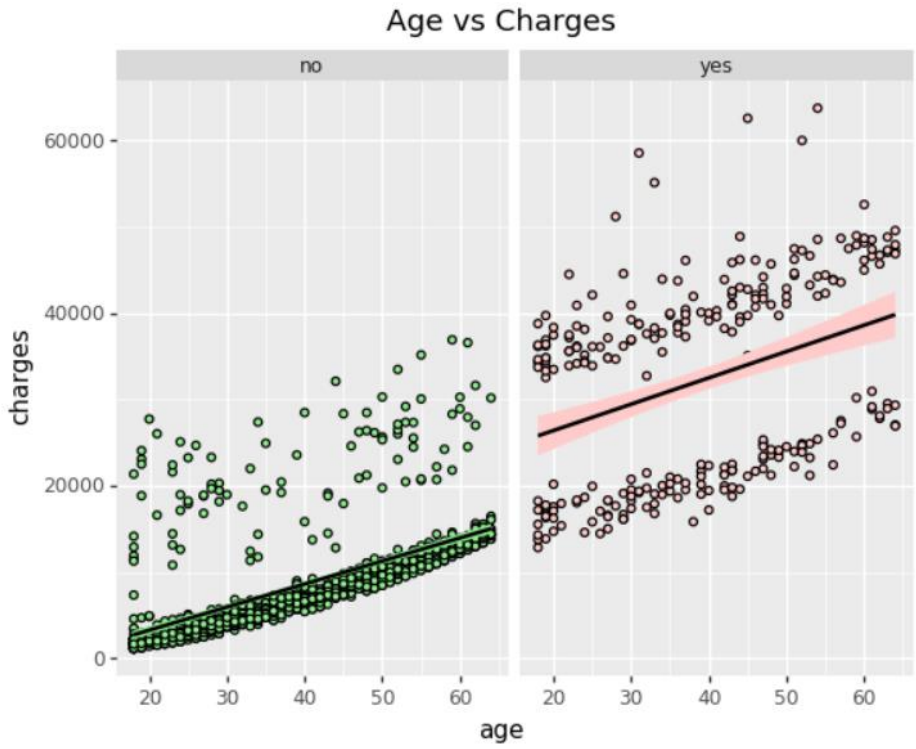


The average price of smokers is considerably much higher than non-smokers. Since smokers generally have a worse state of health and as a consequence the medical charge will be higher.

People with a high BMI the insurance charge is higher



Age influence the price of insurance



We observe 4 "clusters":

- The first is for healthy people who do not smoke are healthy, as a consequence they do not have severe medical problems.
- People who do not smoke but have significant health problems.
- People who smoke but have a good health condition.
- Users who smoke and have serious medical problems.

It can be simplified under two conditions. The first where the condition is not so serious and the second is when the case is dedicated.

We could create an additional feature, to be able to classify users based on the degree of health of the user. Since, as we can see in the graph, the quality of health influences the medical position. Using the histogram and the box plot. We confirm the presence of outliers. So, we have to give it special processing.

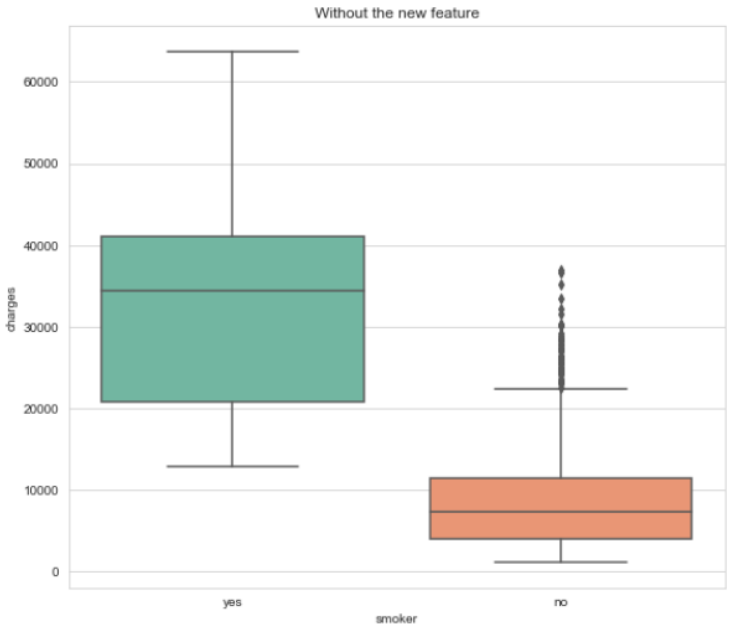
Technically we can give outliers the same treatment as missing values.

- **Delete** those values.
- **Replace** them with a **statistical measure** or by any other value that is in a suitable range.
- **Add** new variables, which we are doing here.

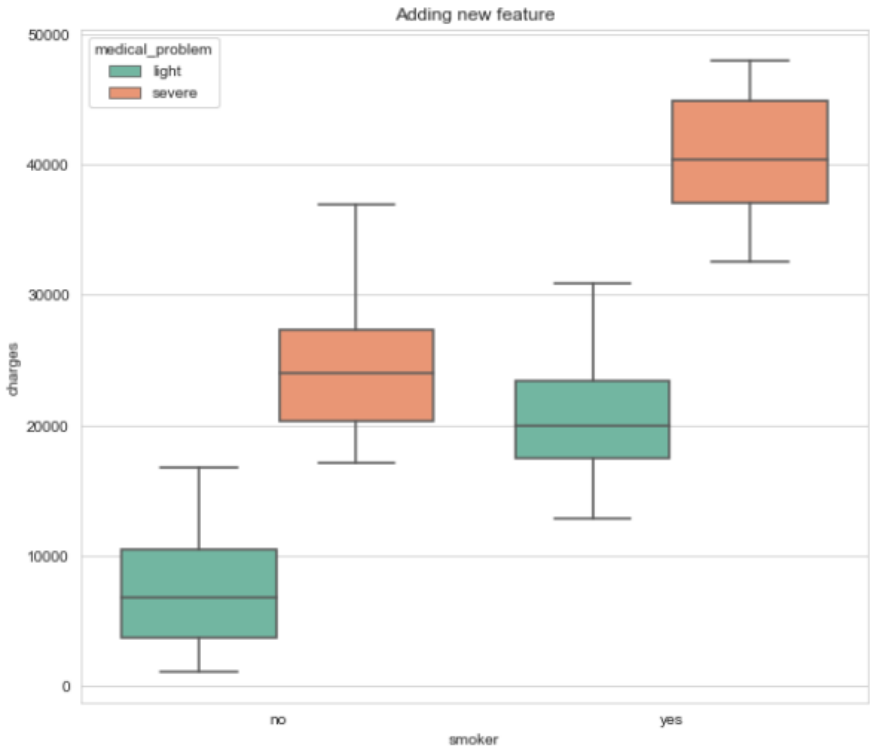
Treatment of outliers

We apply the central limit theorem, which is a technique used to establish confidence intervals. We use these intervals to create a new important feature, which can explain the outliers.

We apply intervals to group based on the cost of health insurance. We divide them into people with less serious medical problems and people with severe medical problems.



After adding new variable (column field) of medical problem



Data Preprocessing

We will only perform **One Hot Encoding transformation** for categorical variable.

One Hot Encoding

id	color
1	red
2	blue
3	green
4	blue

One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

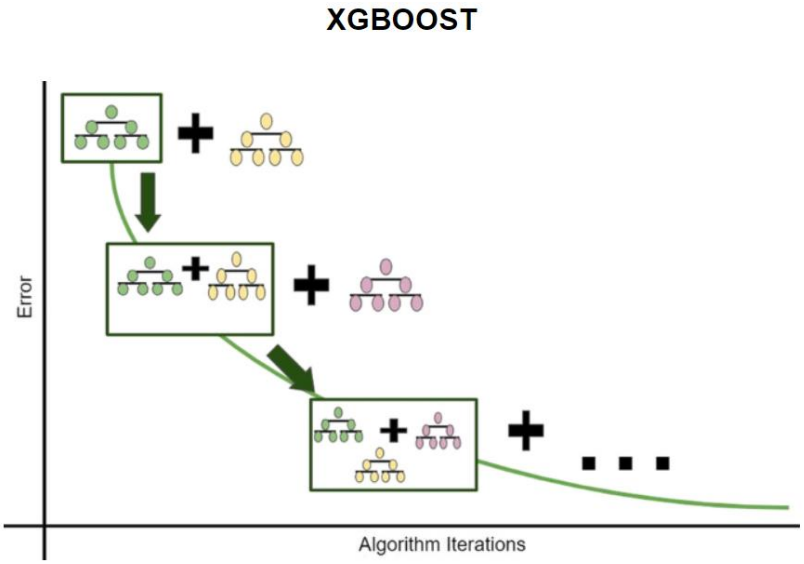
It is used for qualitative categorical variables, for example in the image the color variable. Where dummy variables are created according to the number of categories of the variables, a 1 is assigned where it complies with the condition and the others are filled with 0.

Instead of just replacing the labels with random numbers it can affect the performance of the model, for this type of variables. Since we would be giving more weight to the categories that have the highest value. In addition, the One Hot Encoding transformation has the advantage at the geometric level, since there is already the same distance between the categories.

After preprocessing our test and train data

which you can see in code (jupyter notebook)

We are using XGBOOST here



It is part of the assembly algorithms. Which is a type of algorithms that uses weaker models, generally decision trees. The functioning of this model can be summed up with the following phrase: **"Unity is strength".**

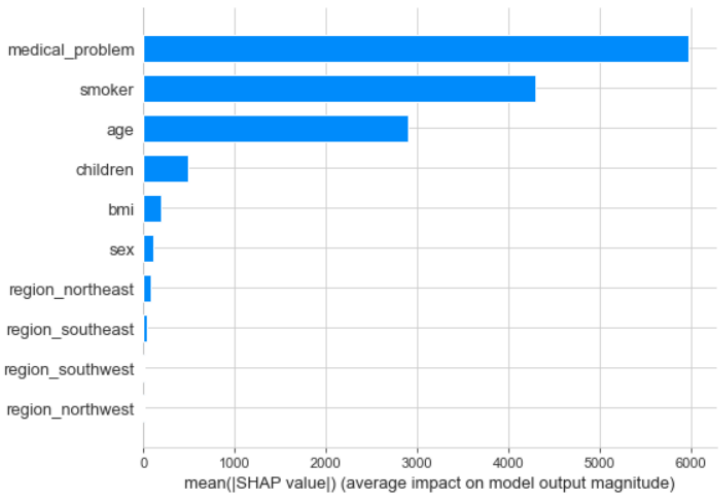
Unlike its brother the random forest which is another ensemble algorithm, it will improve each estimator in such a way that each estimator becomes better than the previous one according to the learning rate.

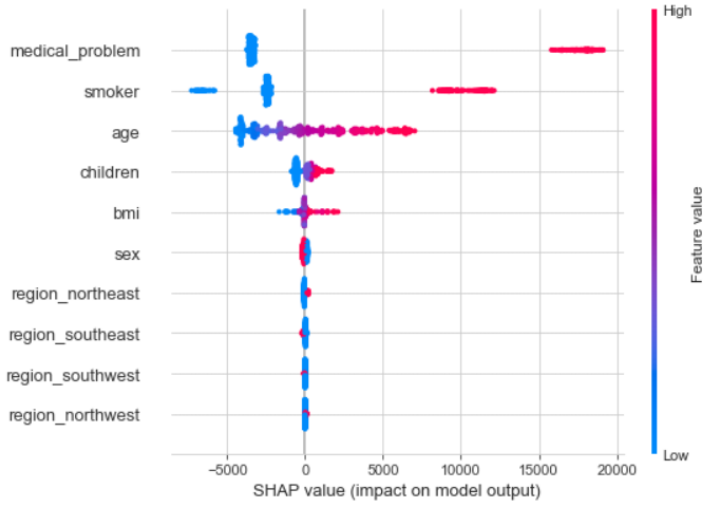
For this particular problem, which is regression, that is, to predict values with a decimal, each estimator performs the prediction to subsequently obtain the average prediction for each estimator.

Explanation of parameters

- max_depth: Maximum depth of each decision tree.
- n_estimators: Number of estimators, that is base algorithms.
- learning_rate: Room for improvement for each decision tree, this parameter goes from 0 to 1.
- random_state: For example, if I want to run this algorithm again, it will give me a different result, due to the random state

Plot Importance





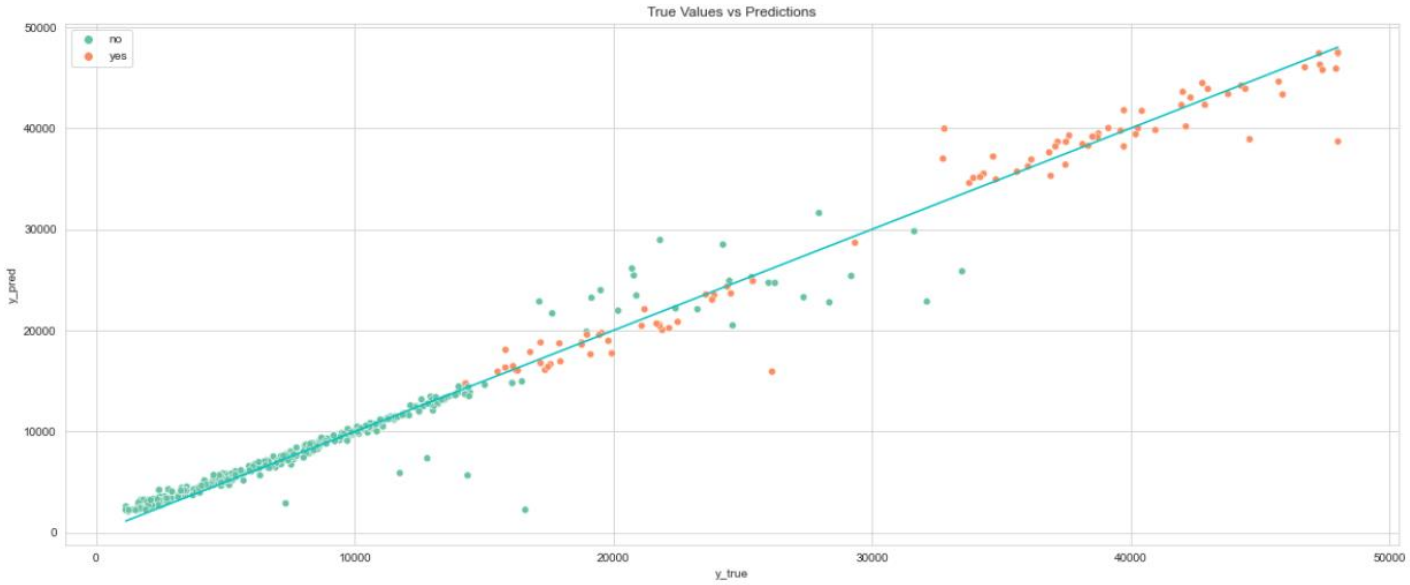
- We note that the **medical problem** variable that we created before. It has great weight when estimating the price of the insurance, since if we have a very serious problem, the cost of the insurance will not increase more.
- The **smoker** variable also has great weight, since people generally have a worse state of health.
- The variable **age** adds value to the predictions. Since it can be understood that elderly people require more medical care.

The other variables may not have as much relevance compared to the previous variables that I mentioned earlier. But they can complement the value of the prediction. And that the difference between humans and machines when making predictions is that we rely on only relevant variables, while machines use these variables and also those that are not so significant, since they look for patterns unknown to the naked eye.

Predictions

The algorithm generates quite robust predictions, ***very close to the original value.*** Which this model is apt to solve the problem.

	y_true	y_pred	smoker
764	11534.87265	11516.696289	no
887	8605.36150	8715.177734	no
890	2396.09590	3481.359863	no
1293	36898.73308	35282.058594	yes
259	18955.22017	19870.119141	no



As a curious fact, XGBOOST is one of the most powerful algorithms within Machine Learning, it generate interesting results in such a short time. It is the winner of multiple competitions on the kaggle platform. It has the advantage that we can use a GPU for training, speeding up the training process, something it shares with Deep Learning frameworks.