Classification .

scikit-learn Machine Learning in Python has Simple and efficient tools for predictive data analysis .

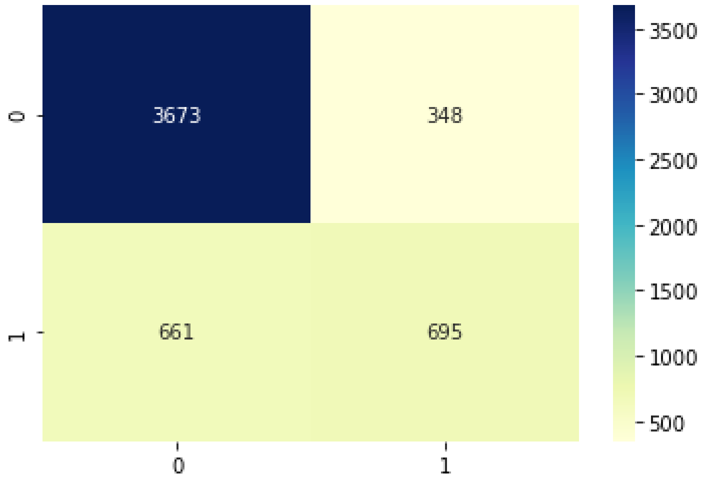
we used for the classification:-

- LabelEncoder from sklearn.preprocessing package to normalize labels such that they contain only values between 0 and n.
- train_test_split from sklearn.model_selection package to split the data into train set and test set.
- RandomForestClassifier from sklearn.ensemble package to improve the predictive accuracy and control over-fitting.
- make_classification from sklearn.datasets package to generate a random n-class classification problem.
- metrics from sklearn package to make a scorer from a performance metric.

At the beginning of the classification phase we transformed some data into numeric intervals then removed unwanted columns and started to split data into x “independent variable” is a 2D array and y “ dependent variable” is a 1D array and then made the train set and test set.

Then we calculated the test accuracy and train accuracy and there is not large difference in accuracy between the test and train , accuracy has no overfitting.

We used the confusion matrix to summary the prediction results on a classification problem then used heatmap to visualize the classifier results.



Confusion Matrix:-

- False Positive = 348 incorrectly predicted event values.
- False Negative = 661 incorrectly predicted no-event values.
- True Negative = 695 correctly predicted no-event values.
- True Positive = 3673 correctly predicted event values.