CMPT459

Milestone2: Report

Mohammad Raad Sarar 301326232
Kazi Dhrubo 301325924
Ahmed Irteza Haque 301325980

Splitting dataset

Training data was further split into train and validation datasets with a train to validation ratio of 80:20. Scikit learn's *train_test_split()* was used to complete this task as follows:

```
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.2, random_state=1)
```

Model Choice

The models chosen are: ***K-nearest Neighbors, AdaBoost,*** *and* ***Random Forest***

Models Not Chosen:

Naive Bayes was not chosen as it assumes that attributes are conditionally independent given a class. But in our dataset many attributes are dependent on each other. For example (longitude and latitude) and (Confirmed, Recovered, Active, Incidence Rate, Case-Fatality Ratio).

Decision Tree was not chosen as Random forest is a better estimator of the decision tree.

SVM assumes a linear plane separating the datapoints. We believe that a dataset of 367635 datapoints with 14 attributes would be quite difficult to separate using a linear decision surface unless the added complexity of the kernel method is implemented which would reduce the interpretability of our model.

Models Chosen:

K-nearest Neighbors: Since KNN uses a distance function to calculate other data points that are closest to a given point we believe data point that are similar will have the same outcome. Similar locations, age group and sex are likely to have similar outcomes. This hypothesis was also the reason why we assigned weights that are inverse to the distance, i.e. distant objects will have less weight.

Since KNN depends on distances, we scaled the data so that all attributes contribute equally to the distance calculations. Columns province and country were dropped as label encoder would cause inconsistency in distance calculations. This will not be a problem because latitude and longitude columns will serve the purpose of locating geographically nearby points. Also, the sex column is one hot encoded.

Random Forest: Decision tree split attributes that have the highest Information Gain/Gini Index which result in more accurate and sophisticated decision surface. These are desirable properties however, decision trees have high variance and are likely to overfit. Therefore, we use Random Forest Classifier which is a voting-based classifier that reduces the variance and results in more comprehensive classifications.

AdaBoost: We know that decision trees have some desirable properties as mentioned above. AdaBoost uses Decision Trees as its base classifier and fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. https://scikit-learn.org/

For all classifiers, the date_confirmed column was converted to hours from January 1st, 2020 so that the column contained numerical data.

For Random forest and AdaBoost, categorical columns (Sex, Province and Country) were labeled using sklearn's label encoder as these classifiers do not rely on distances.

Evaluation and Overfitting:

Evaluation was done while parameter tuning to detect overfitting.

We varied hyperparameters for each classifier and calculated training and validation accuracies. Then we plotted a graph of train/validation accuracies against the hyperparameter values. For different hyperparameter values we also recorded the precision, recall and confusion matrix for train and validation datasets.

n_neighbors	training_score	validation_score	training_confusion_matrix	training_precision	training_recall	validation_confusion_matrix	validation_precision	validation_recall
1	0.911348591	0.769404436	[[2376 748 12 493]	0.835756485	0.838710815	[[64 443 68 295]	0.574420504	0.575425938
2	0.912079617	0.773552573	[[2888 608 1 132]	0.823500236	0.865082802	[[86 489 68 227]	0.577895807	0.577550357
3	0.921980776	0.779196758	[[2660 667 2 300]	0.837466771	0.864389707	[[67 456 64 283]	0.582930396	0.582450081
4	0.924160255	0.784732139	[[2507 793 2 327]	0.877362447	0.85299694	[[57 499 67 247]	0.590969839	0.581425736
5	0.926594743	0.787017014	[[2507 828 2 292]	0.87589169	0.85733971	[[51 483 68 268]	0.591664289	0.583363393

Figure 1: scores for knn using k=1-5

Using this data for each classifier we plotted the following graphs:

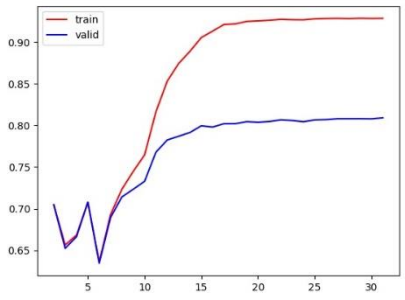

Figure 2: AdaBoost with varying max_depth

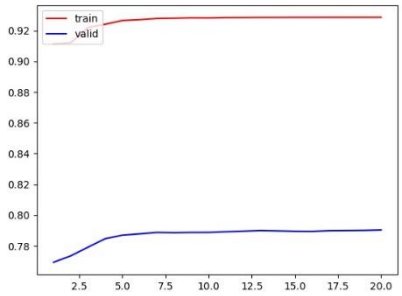

Figure 3: KNN with varying k

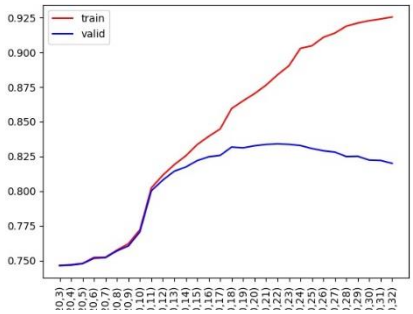

Figure 4: Random Forest with varying max_depth

In our training and validation graphs we looked for points where the validation data stopped increasing or started decreasing to determine the best hyper-parameter. For overfitting detection, we looked for points in the graph where the validation score no longer increases but starts decreasing. For the random forest we observed slight overfitting after max_depth = 19 (figure 4). For KNN we did not observe overfitting (curve flattening after k=7 in figure 3) because we used the inverse of the distance as the weights which means the more distant a neighbor is the lower it's impact on the classification of a point. AdaBoost also did not show overfitting as the curve just flattened after max_depth =16 (Figure 2) the initial spikes are probably caused as the max_depth is too low to classify correctly.

Evaluation Metrics of the best Hyperparameter for each Classifier:

Classifier	Best Hyperparameter	Training Accuracy	Validation Accuracy	Validation Precision	Validation Recall
Random Forest	Max_depth =19	87%	83%	71%	61%
KNN	K= 7	93%	79%	60%	59%
AdaBoost	Max_depth =16	91%	80%	61%	59%

Validation accuracy is an appropriate metric because we can detect overfitting and precision is another one because it gives us the percentage of how many are actually positive among all positive predictions.