Machine learning applied to Cervical Cancer Data

Vineet Menon, Dhwaani Parikh

Abstract— Cervical Cancer is one of the main reason of deaths in countries having a low capita income. It becomes quite complicated while examining a patient on basis of the result obtained from various doctor's preferred test for any automated system to determine if the patient is positive with the cancer. We will try to use machine learning algorithms and determine if the patient has cancer based on numerous factors available in the dataset. Predicting the presence of cervical cancer can help the diagnosis process to start at an earlier stage.

Index Terms— Cancer, Cervical cancer, Decision tree Classifier, herpes virus, K-nearest neighbor, Machine learning, Random forest

1) Introduction

Cervical Cancer is cancer arising from the cervix. It arises due to the abnormal growth of cells and spreads to other parts of the body. It is fatal most of the time. HPV causes most of the cases (90 %). In Phase I, the data is cleaned, and visualisations of the data are shown. Smoking is also considered as one of the main causes for cervical cancer. Long-term use of Oral contraceptive pills can also cause cancer. Also having multiple pregnancies can cause cervical cancer. Usually it is very difficult to identify cancer at early stages. The early stages of cancer are completely free of symptoms. It is only during the later stages of cancer that symptoms appear. We can use machine learning techniques to predict if a person as cancer or not. Different factors such as smoking, pregnancies, habits etc can be used to predict cancer.
In phase 1 data is cleaned and visualisations are shown. In Phase II, the methodology is shown. Different models are tried on the dataset. Fine tuning of the parameters is done. The performance of different models is compared. And finally, the best models are recommended that can be used to predict cancer.

2) Methodology

For modelling three different classifier techniques are used. Decision tree, K-nearest neighbour and random forest. From the phase 1 it is found that the dataset is biased
hence k-fold-cross-validation is done. For k-nearest neighbour dataset is split 50-50. Feature selection is done, and the selected features are used for prediction. For decision tree and random forest, the data is split 25-75. Parameter tuning is done to get the best predictions with optimal evaluation scores. We also used Hill climbing Algorithm for feature selection. The algorithm is a mathematical optimization that use heuristic search in the felid of Artificial Intelligence. Following are steps how the algorithm does a feature selection.

Step1: -Make initial state as current state after evaluating the initial state

Step2: -Run the loop till there are no features present which can be applied to current state.

a) Select a feature that has not been yet applied to the current state and apply it to produce a new state.
b) Perform these to evaluate new state
   i. If the current state is a goal state, then stop and return success.
   ii. If it is better than the current state, then make it current state and proceed further.
   iii. If it is not better than the current state, then continue in the loop until a solution is found.

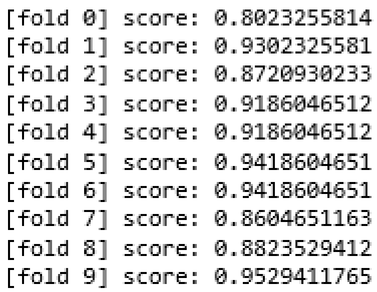
Step3: - Exit

Confusion matrix, classification error rate, Recall, F1score are used to evaluate different models. AUC curve is used to evaluate the model and is used to select the best model by parameter tuning.

2.1)K-Nearest Neighbour Classifier:

K-fold cross validation:

For the nearest neighbour K-fold cross validation is used. Appropriate k value is selected based on the formula sqrt(N)/2 which gives us 10.3. The sample size of the training dataset is chosen to be 429.After running several models the value of K is chosen as 5. Euclidean distance method is used to calculate the distance between two values.
The result for the k-fold cross validation technique is shown below.



```
[fold 0] score: 0.8023255814
[fold 1] score: 0.9302325581
[fold 2] score: 0.8720930233
[fold 3] score: 0.9186046512
[fold 4] score: 0.9186046512
[fold 5] score: 0.9418604651
[fold 6] score: 0.9418604651
[fold 7] score: 0.8604651163
[fold 8] score: 0.8823529412
[fold 9] score: 0.9529411765
```

Fig 1: K-fold cross validation

Lower value of k is biased. Higher values of k is less biased but it can show variance. Since k=5 which is neither less nor more.
For the k-nearest neighbour the dataset is split 50-50. Feature selection is done which help us to select the features that will improve the prediction. The 25 features selected for the predictions are

Features selected for K-Nearest Neighbour Classifier	
Age	STDs (number)
Number of sexual partners	STDs:condylomatosis
First sexual intercourse	STDs:cervical condylomatosis
Num of pregnancies	STDs:vaginal condylomatosis
Smokes	STDs:vulvo-perineal condylomatosis
Smokes (years)	STDs:syphilis
Smokes (packs/year)	STDs:pelvic inflammatory disease
Hormonal Contraceptives	STDs:genital herpes
Hormonal Contraceptives (years)	STDs:molluscum contagiosum
IUD	STDs:AIDS
IUD (years)	STDs:HIV
STDs	STDs:Hepatitis B
	STDs:HPV

Table 1: Features selected for K-Nearest Neighbour Classifier

2.2)Decision tree Classifier:

For the decision tree classifier, the dataset is split 25%. Presort function is used for fast implication of the algorithm. To minimize the size of the tree the minimum split for sample is set at 110.Class_weight is used as balanced so that it automatically adjusts the weight inversely proportional to class frequencies in the input data.
Feature selection is done for decision tree classifier. The following features are selected for the prediction

Features selected for Decision tree Classifier	
STDs:AIDS	Dx:CIN
STDs:herpes	STDs:vaginalcondylomatosis
STDs:cervicalcondylomatosis	Dx
STDs:HPV	Dx:HPV
STDs: Time since first diagnosis	STDs:AIDS
Smokes	STDs:vulvo-perineal condylomatosis
First sexual intercourse	STDs:syphilis
IUD (years)	STDs:syphilis
	STDs:contagiosum

Table 2: Features selected for Decision tree Classifier

2.3) Random Forest algorithm

The random forest algorithm uses the training data set and generates multiple level decision tree. For the decision tree the data is split 25-75 for training and testing data. The depth of the tree is limited to 10 to make the tree less complex. After running the algorithm several times, the maximum sample split is decided to be 75. As per the inverse of frequency of input data class weight is again used as 'balanced' for automatic adjustment.
For the random forest after we do the feature selection only 11 features are selected for prediction. The features are

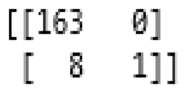
Features selected for Random Forest	
STDs:HPV	Smokes
STDs:molluscum contagiosum	First sexual intercourse
STDs: Time since first diagnosis	IUD (years)
IUD	Dx:CIN
Dx	STDs:vaginal condylomatosis
	Dx:HPV

Table 3: Features selected for Random Forest.

3) Evaluation

3.1) Confusion Matrix:

K-nearest neighbour classifier:



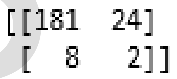
```
[[163   0]
 [  8   1]]
```

Fig 2: Confusion matrix of K-nearest neighbour classifier.

According to the above confusion matrix, [1]

- True positive count is 163.
- False negative count is 0.
- False positive count is 8.
- True negative count is 1.

Decision Tree Classifier:



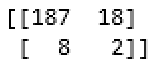
```
[[181  24]
 [  8   2]]
```

Fig 3: Confusion matrix of Decision tree classifier

According to the confusion matrix, [1]

- True positive count is 181.
- False negative count is 24.
- False positive count is 8.
- True negative count is 2.

Random Forest classifier



```
[[187  18]
 [  8   2]]
```

Fig 4: Confusion matrix of Random forest classifier

According to the confusion matrix, [1]

- True positive count is 187.
- False negative count is 18.
- False positive count is 8.
- True negative count is 2.

3.2) Accuracy

Accuracy of the model is given by the formula [1]

Accuracy =(TP+TN)/total

K-nearest neighbour: Accuracy of this algorithm is **95.3%.**

Decision Tree Classifier: Accuracy of this algorithm is **85.11%.**

Random-Forest: Accuracy of this algorithm is **87.90%**

3.3) Classification Error Rate

Classification error rate is given by the formula:

CER = 1-accuracy

K-nearest neighbour: The classification error rate is found out to be **4.7%.**

Decision Tree Classifier: The classification error rate is found out to be **14.89%.**

Random-Forest: The classification error rate is found out to be **12.1%**

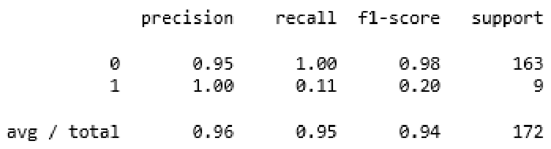
3.4) Precision Recall and F1 Score

Precision = TP/TP+FP

Recall = TP/TP+FN

F1 Score = 2(Recall * Precision) / (Recall + Precision)*

K-nearest neighbour:



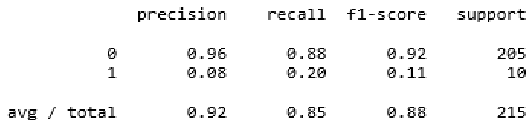
```
             precision    recall  f1-score   support

          0       0.95      1.00      0.98       163
          1       1.00      0.11      0.20         9

avg / total       0.96      0.95      0.94       172
```

Fig 5: Precision Recall and F1 Score of K-nearest neighbour classifier

Decision Tree Classifier:



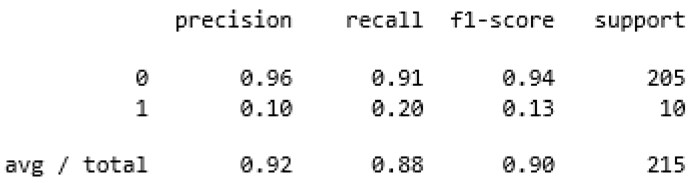
```
             precision    recall  f1-score   support

          0       0.96      0.88      0.92       205
          1       0.08      0.20      0.11        10

avg / total       0.92      0.85      0.88       215
```

Fig 6: Precision Recall and F1 Score of Decision tree classifier

Random Forest Classifier:



```
             precision    recall  f1-score   support

          0       0.96      0.91      0.94       205
          1       0.10      0.20      0.13        10

avg / total       0.92      0.88      0.90       215
```

Fig 7: Precision Recall and F1 Score of Decision tree classifier

3.5) AUC/ROC

K-nearest neighbour:

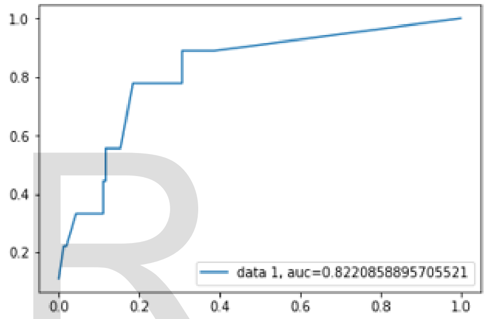


Fig 8: AUC/ROC of K-nearest neighbour classifier

The figure shows the AUC chart and the AUC value is good which is 0.8220. This model can be used for prediction

Decision Tree Classifier:

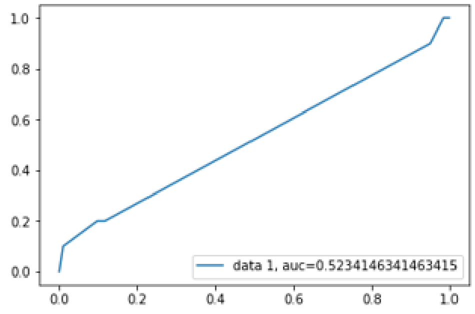


Fig 9: AUC/ROC off decision tree classifier

The AUC value for Decision tree is 0.52. It is very less compared to K-nearest neighbour method. But this AUC value is acceptable since it is more than 0.5.

Random-Forest

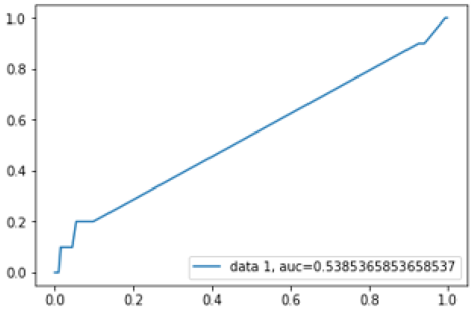
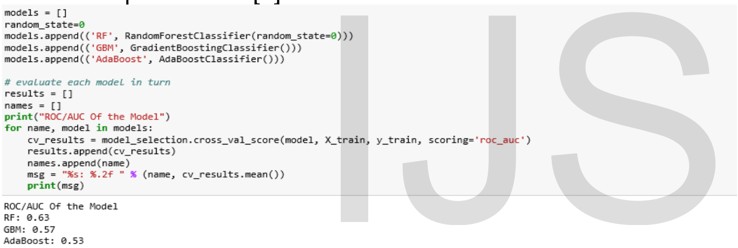


Fig 10: AUC/ROC off decision tree classifier

The AUC curve for random forest is similar as decision tree with an AUC value of 0.538536

4) Why Select Random Forest?

Comparison is also made based on the AUC and ROC of a model to decide which model to implement on the given data for correct prediction. [2]



```
models = []
random_state=0
models.append(('RF', RandomForestClassifier(random_state=0)))
models.append(('GBM', GradientBoostingClassifier()))
models.append(('AdaBoost', AdaBoostClassifier()))

# evaluate each model in turn
results = []
names = []
print("ROC/AUC Of the Model")
for name, model in models:
    cv_results = model_selection.cross_val_score(model, X_train, y_train, scoring='roc_auc')
    results.append(cv_results)
    names.append(name)
    msg = "%s: %.2f " % (name, cv_results.mean())
    print(msg)
```

```
ROC/AUC Of the Model
RF: 0.63
GBM: 0.57
AdaBoost: 0.53
```

Fig 11: Code chunk and result for choosing Random forest

Different models such as Random forest, Gradient Boosting classifier and AdaBoost. The results in the above figure prove that random forest has the best ROC/AUC than the other two models. Hence, we chose to go ahead with random forest classifier.

5) Discussion

After completing the analysis, it is found that all the models are good that is k-nearest neighbour, decision tree and random forest. K-nearest neighbour seems to be the best model with higher accuracy of the model, Higher AUC which is 0.822 as compared 0.52(Decision tree) and 0.532(random forest) which is very low. Precision recall and f1 score is also high for nearest-neighbour model. The f1 score for nearest-neighbour is 0.94 which is high when compared to decision tree (0.88) and random forest (0.90).

From the confusion matrix we can also note that false negative is zero which means that a patient with cancer will have prediction that he does not have cancer. It will be a bad scenario when a person having cancer will be informed that he does not have cancer. By the time symptoms starts showing it could be too late. So, it is important to choose a model that has very low false negative rates in some cases such as ours. Also, the k-nearest neighbour model has the best accuracy and AUC value which is one of the strengths of the model. So, k-nearest neighbour will be used for prediction.

The dataset is biased it has a lot of zero values i.e. the patient does not have cancer for the target variable. If the dataset was little less-biased, we would have got better models and more accurate predictions. Simple climbing technique used has some disadvantages. It is difficult to know if the hill found is the highest possible hill. The local maxima are a state that is best for its neighbouring states, but it might not be better than the states further away.

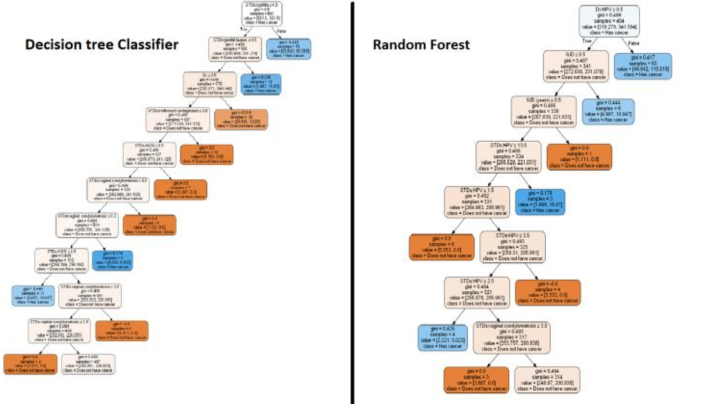


Fig 12: Tree created after pruning by Decision tree and Random Forest.

6) Conclusion

- After the dataset is cleaned in the phase1 we have proceeded with the modelling part for the phase2.
- From phase 1 it is found that our dataset is biased.
- k-fold cross validation is used for modelling.
- Three types of models are used k-nearest neighbour, decision tree and random forest.
- Fine tuning is done on all three models to get the best accuracy.
- The best 3 models from each of them is selected and performance is compared
- It is found that all 3 models are good.
- But the k-nearest-neighbour model has better accuracy, precision, recall and better AUC value.
- The research also showed that herpes virus was able to fight cancer cells this observation is was made based on the data available and more scientific analysis needs to be carried out in order to confirm this observation.

7) References

[1] R), I & R), I 2018, "Improve Your Model Performance using Cross Validation (in Python / R)", *Analytics Vidhya*, viewed 14 May, 2018, https://www.analyticsvidhya.com/blog/2018/05/improve-model-performance-cross-validation-in-python-r/

[2] Anon 2018, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog", *Exsilio Blog*, viewed 26 May, 2018, <http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>.

[3] Anon 2018, *S3.amazonaws.com*, viewed 26 May, 2018, https://s3.amazonaws.com/MLMastery/MachineLearningAlgorithms.png?__s=8cjo2uzaybyqe5a1fipt