DATA MINING & WAREHOUSING

DEVANSHI VATS
BTECH CSE
2218680
C2
5TH SEM

Table of Contents

Lung Cancer Prediction Model

Abstract:

Our project focuses on building a program which predicts lung cancer in any patient by measuring the attributes of the collected data like the frequency of smokes, age, air quality index of the area where patient resides, the consumption of alcohol by the patient and dietary data. The main driving force for conducting this research is that American Cancer Society's estimates the figure for United States for 2022 are, about 236,740 new cases of lung cancer (117,910 in men and 118,830 in women). Moreover, there are about 130,180 deaths from lung cancer (68,820 in men and 61,360 in women). Not to mention we have recorded those states where there were an estimated 2.1 million lung cancer cases and 1.8 million deaths in 2018 globally.[1] This program would help reduce the number of lung cancer patients. It can help save someone's life.

We have used three algorithms for getting the best results, this would include KNN, and Naïve Baye.

We start off by data cleaning whereby we check for missing data. We find all the data fields filled completely so there is no missing data. Name and surname attributes contribute to noisy data as they are not co-related with other attributes, therefore we use data reduction to remove these attributes. We have used attribute subset selection here as the highly relevant attributes are used, while the rest are discarded. For performing attribute selection, we use level of significance and p- value of the attribute. The attribute having p-value greater than significance level are discarded. Then we move to data transformation where we normalize result attribute. The result attribute is already normalized between the range 0 and 1. As we have removed two dimensions from our data set as it does not correlate with other dimensions. After preprocessing we applied the said classifiers where we classify things into sub-categories. We are using binary classification as it will predict whether the patient has cancer or not.

Literature Review:

In the first article, we see the researcher talking about the risk model for prediction of lung cancer.

Reliable risk prediction systems for evaluating individual lung cancer risk have significant public health implications. We developed and validated a discursive clinical method for predicting lung cancer risk based on the degree of smoking.

Epidemiologic data from 1851 lung cancer patients and 2001 matched control subjects were randomly classified into training (75 percent of the data) and validation (25 percent of the data) groups for never, ex-, and current smokers, and multivariable models were built using the training sets. In the validation sets, the model's discriminatory tendency was evaluated by the areas under the receiver operating characteristic curves coupled with concordance statistics. Using national incidence and mortality demographics, precisely 1-year risks of lung cancer were calculated. By taking the summation of odds ratios from the multivariable regression models for each risk factor we were able to create an ordinal risk index for each smoking level.

Environmental tobacco smoke, family history of cancer, dust exposure, prior respiratory illness, and smoking history are all factors that have a statistically significant connection with lung cancer. In the validation sets, the concordance statistics for the never, former, and current smoker models were 0.57, 0.63, and 0.58, respectively. The calculated 1-year definite risk of lung cancer for an assumed male current smoker with a relative risk close to 9 was 8.68 percent. In the true-positive rates, ordinal risk index, performed well in the specified high-risk groups for current and past smokers were 69 percent and 70 percent, respectively. If verified in future research, this risk assessment approach might employ easily acquired clinical data to identify patients who may benefit from long screening surveillance for lung cancer. Although the concordance statistics were fair, they are harmonious with further risk prediction models.

In our second article we discover deep learning approach to predicting lung cancer.

Deep Learning has been offered as a potentially useful approach for classifying cancerous nodules. Our goal was to verify our Lung Cancer Prediction Convolutional Neural Network (LCP-CNN), which was trained on US screening data, using an independent dataset of indeterminate nodules in a European multicenter experiment, in order to rule out benign nodules while preserving high lung cancer susceptibility. The LCP-CNN was trained to produce a malignancy rating for each nodule using CT data from the United

States National Lung Screening Trial (NLST) and confirmed on CT scans containing 2106 nodules meaning 205 lung cancers which were detected in patients from the Early Lung Cancer Diagnosis Using Artificial Intelligence and Big Data (LUCINDA) study, selected from three tertiary referral points in the United Kingdom, Germany, and the Netherlands. By computing limits on the malignancy score that reach at least 99 percent responsivity on the NLST data, we pre-defined a benign nodule rule-out test to detect benign nodules while retaining a high sensitivity. The overall performance of each validation site was assessed using the Area-Under-the-ROC-Curve technique (AUC). The total AUC across European centers was 94.5 percent (95 percent confidence interval 92.6–96.1). Malignancy could be ruled out in 22.1 percent of the nodules with a high sensitivity of 99.0 percent, allowing 18.5 percent of the patients to skip follow-up scans. Both false-negative results were for little common carcinoids.

Preprocessing:

We applied different checks to identify the noisy and inconsistent data present in the data set. But we observed that there was no noisy data present, so we did not perform any preprocessing on the data set. Also, the data set was pretty observable as there were only 60 entities.

Datatypes:

```
[6]  data.dtypes

Age       int64
Smokes    int64
AreaQ     int64
Alkhol    int64
Result    int64
dtype: object
```

Here, we have 7 columns in our dataset. The first two columns are Name and Surname which are of no use. So, we simply dropped these two features.

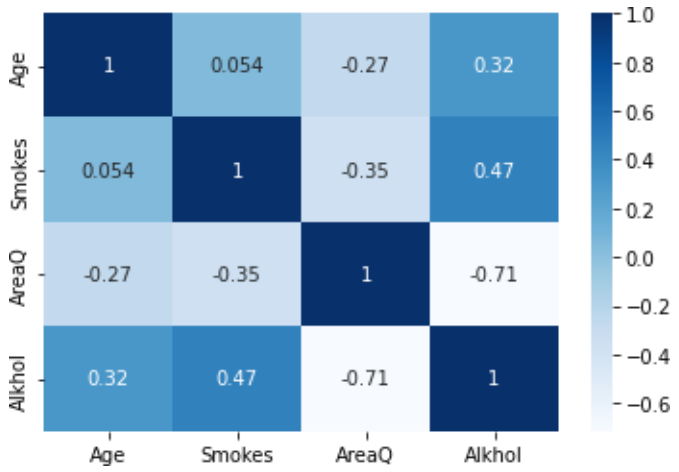
Dropping Name and Surname Columns

```
[4]  data = data.drop(['Name', 'Surname'], axis=1)
     data.head()
```

	Age	Smokes	AreaQ	Alkhol	Result
0	35	3	5	4	1
1	27	20	2	5	1
2	30	0	5	2	0
3	28	0	8	1	0
4	68	4	5	6	1

The other features are; age of the patient, is the patient is smoker or not also here we are given the frequency of smoke, how many cigarettes a patient consumes in a day, and the quality of air where the patient is living, we are given the index of air quality. Moreover, another feature is alcohol consumption, which demonstrate how a person is frequent in alcohol drinking.

We find correlation of the features, to get the relation of our input features. And we observed that almost every feature is somehow corelated with each other. And air quality index and alcohol are negatively but highly corelated.




```
[7]  data.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 59 entries, 0 to 58
     Data columns (total 5 columns):
      #   Column  Non-Null Count  Dtype
     ---  ------  --------------  -----
      0   Age     59 non-null     int64
      1   Smokes  59 non-null     int64
      2   AreaQ   59 non-null     int64
      3   Alkhol  59 non-null     int64
      4   Result  59 non-null     int64
     dtypes: int64(5)
     memory usage: 2.4 KB
```

Naïve Bayes:

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. works on the principle of conditional probability, as given by the Bayes theorem. While calculating the math on probability.
We split the data by using the test size of 0.2 meaning we used 80% of the data for training the model and 20% of the data for testing the model. We used Skit learn library to import and implement Gaussian Naive bayes model on the data.

- We fit the training data into the model and
- we then made the predictions by using the test data.
- We have achieved accuracy of 92%.
- We have used several evaluation techniques to measure accuracy of the system like Mean Squared Error which is 8.3% for our model. We then used confusion matrix and classification report which is attached as follows

```
Report:
              precision    recall  f1-score   support

           0       0.88      1.00      0.93         7
           1       1.00      0.80      0.89         5

    accuracy                           0.92        12
   macro avg       0.94      0.90      0.91        12
weighted avg       0.93      0.92      0.91        12
```

Confusion Matrix:

```
[177] confusion_matrix(y_test,y_predict)

array([[7, 0],
       [1, 4]])
```

Logistic Regression:

Second model is Logistic Regression which is proven to be most suitable to classify when we have binary attributes and binary target variable and in our case our target variable is binary. We have now used Logistic Regression which is proven to be suitable to classify when we have binary attributes and binary target variable and in our case our target variable is binary.

We split the data by using the test size of 0.4 meaning we used 60% of the data for training the model and 40% of the data for testing. We used Skit Learn library to import and implement Logistic regression model. We fit the training data into the model, and we then made the predictions by using the test data. An accuracy of 96% is achieved. Mean Squared Error which is 4.16%. We then used confusion matrix and classification report which is attached as follows:

Report:

Report:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	10
1	1.00	0.93	0.96	14
accuracy			0.96	24
macro avg	0.95	0.96	0.96	24
weighted avg	0.96	0.96	0.96	24

Confusion matrix:

```
[178] confusion_matrix(y2_test,y2_predict)

array([[10,  0],
       [ 1, 13]])
```

K- Nearest Neighbor:

We divided the data by using a test size of 0.40, which meant that we used 60 percent of the data to train the model and 40 percent of the data to test the model. To import and implement the KNN model on the data, we used the Skit learn library. The value of k was set to 5. (Nearest neighbor). To achieve better results, use an odd-numbered k. We used the test data to make predictions after fitting the training data into the model. A 64.5 percent accuracy was obtained. We used several evaluation techniques to determine the system's accuracy, such as Mean Squared Error, which is 0.355. The confusion matrix, which is attached, was then used.

Confusion matrix:

```
[200] confusion_matrix(y3_test,knn_predict)

      array([[10,  2],
             [ 0, 12]])
```

Report:

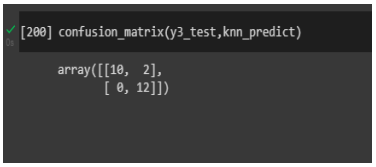
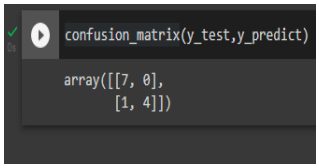
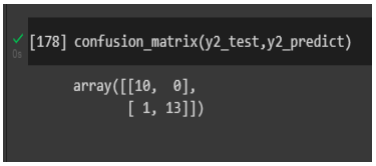
```
[202] print("Report: ")
      knn_report =classification_report(y3_test,y3_predict)
      print (knn_report)

      Report:
                    precision    recall  f1-score   support

                 0       1.00      0.83      0.91        12
                 1       0.86      1.00      0.92        12

          accuracy                           0.92        24
         macro avg       0.93      0.92      0.92        24
      weighted avg       0.93      0.92      0.92        24
```

Comparison of all applied algorithms:

Techniques	K-NN	Naïve Bayes	Logistic Regression
Accuracy	92%	91%	95%
Confusion matrix			
Mean-squared error	0.0833	0.0833	0.0416
F1-Score	0.91	0.92	0.96

Conclusion:

As we conclusively proved this article, we read a lot of articles, and each article has its own set of weaknesses and advantages for using dataset techniques. However, after reading all of the articles, we concluded that cardiovascular attacks are increasing day by day, and the number of deaths caused by this is significant. We developed the best program by incorporating all of the positive aspects of these studies and experiments. We examined the data set that they used and developed a more accurate and precise lungcancer prediction model. Logistic Regression proved to be the most stable machinelearning model for this data, as it provided the highest accuracy. However, the naïve bayesand K-NN algorithms are also working pretty well.

Future Work:

As future work some sort of limitations was applied to the work is that different types of classifiers added here to be more in-depth sensitive analysis can be held by help of these classifiers, more in that this analysis can be applying to another bioinformatics dataset and see the performance of these classifiers to classify or predict the diseases.

References:

1. Hong, W. K., Amos, C. I., Wu, X., Schabath, M. B., Dong, Q., Shete, S., & Etzel, C. J. (2007). A Risk Model for Prediction of Lung Cancer. *JNCI Journal of the National Cancer Institute*, *99*(9), 715–726. https://doi.org/10.1093/jnci/djk153
2. Heuvelmans, M. A., van Ooijen, P. M., Ather, S., Silva, C. F., Han, D., Heussel, C. P., Hickes, W., Kauczor, H. U., Novotny, P., Peschl, H., Rook, M., Rubtsov, R., von Stackelberg, O., Tsakok, M. T., Arteta, C., Declerck, J., Kadir, T., Pickup, L., Gleeson, F., & Oudkerk, M. (2021). Lung cancer prediction by Deep Learning to identify benign lung nodules. *Lung Cancer*, *154*, 1–4.
3. https://doi.org/10.1016/j.lungcan.2021.01.027
4. https://towardsdatascience.com/why-training-set-should-always-be-smaller-than-test-set-61f087ed203c
5. https://www.python-graph-gallery.com/92-control-color-in-seaborn-heatmaps
6. https://machinelearningmastery.com/make-predictions-scikit-learn/
7. https://www.bitdegree.org/learn/train-test-split
8. https://python-course.eu/machine-learning/naive-bayes-classifier-with-scikit.php