*Detection of Ovarian Cancer Using Semi Supervised Learning*

Pradyumna Anil Kumar Kubear

*Department of Engineering And Computer Science*  
Universiy Of The PacificStockton CA,USA

p\_anilkumarkubear@u.pacific.edu

*Abstract*— Ovarian cancer is prevailing type of cancer seen in women. The main intention of this project is to detect ovarian cancer using a novel approach of semi-supervised learning.

Keywords—Ovarian Cancer, semi supervised learning, deductive learning, inductive learning

# Introduction

Early detection and appropriate diagnosis of cancer have become increasingly important for decreasing death rates.

If this cancer is not treated properly, it can spread and cause further complications, so to detect this cancer early, several researchers are struggling to come up with an appropriate algorithm.

The data for this study was downloaded from the Kaggle website.

Kaggle is the largest community group that has datasets available for data science projects.

The dataset for this project primarily consists of 51 attributes, only 10 of which are assessed for accuracy, and Shapley's model has revealed just 10 major causes.

The main problem is that we do not get a dataset that is complete and labelled.

In order to address these issues, we scrutinized the use of inductive and deductive learning methods in this study.

The proposed solution incorporates a decision tree classifier for inductive learning and the rule fit algorithm for deductive learning.

The ten traits are: HE4 (Human Epididymis Protein 4), NEU (Neutrophil), ALB (Albumin), Age, GLO (Globulin), IBIL (Indirect Bilirubin), HGB (Haemoglobin), PLT (Platelet), and Menopause.

Another challenge faced was denial from the Practo organization.

When the first implementation using deductive learning was made, the accuracy was very poor; this was also another challenge.

Many attributes were blank, so to make the algorithm more efficient, the right approach for replacing blank spaces had to be found. This was another challenge.

We will address the issues related to limited data in the dataset, and methods that were followed to increase the accuracy of semi-supervised learning throughout this paper.

The ultimate target is to create a feasible model for determining Ovarian Cancer.

# Related work

Initially, comparisons of biomarkers were made, which are as follows: BIL (bilirubin) versus IBIL (indirect bilirubin) vs. TBIL (total bilirubin) [1], an article about biomarkers called red blood cells, white blood cells, and platelets was referred to determine if they help in determining ovarian cancer. After reading this article, it was found out that many people have a high white blood cell count, but for ovarian cancer, this is not a suitable marker [2]. The third research paper spoke about the marker basophil [3]. It is said that basophil is one of the highlighters for accentuating ovarian cancer. Creatine is one of the markers of ovarian cancer [4]. It is said that more the level of creatine more the likelihood of Ovarian Cancer. The neutrophil count is increased in the case of ovarian cancer [5]. It is also found that eosinophils are increased in cases of ovarian cancer [6]. It is also found that gamma glutamyl transferase activity is increased in cases of ovarian cancer [7]. HE4 is a pivotal marker in ovarian cancer. HE4 stands for Human Epididymis Protein 4 [8]. The eosinophil count for ovarian cancer is also high [9].

Menopause also triggers ovarian cancer [10].

The neutrophil count is also high for ovarian cancer [11].

After considering all these attributes, I read a research paper that states to replace zeros for blank and incomplete values in the dataset [12].

After all these, the accuracy was not as expected; it was around 35% to 40%, so in order to increase the accuracy, I used the approach of semi-supervised learning (Shapley’s Plotting) graphs, which comprised attributes like HE4, NEU, CA125, ALB, Age, GLO, IBIL, HGB, PLT, and Menopause [13].

I also read about the implementation of rule-based regression and later thought of implementing the same for my project [19].

# Solution and scalability justification

This project mainly comprises of 2 main approaches which mainly uses semi supervised learning.

For inductive learning we are making use of Decision Tree classifier and for deductive learning we are making use of Rule Based Regression algorithm.

These approaches are chosen because they are easily interpretable, and they are adaptable to varying sizes.

A) Inductive learning using Decision Tree Classifier:

This algorithm was chosen because it splits the training data into more features, it generates a tree like structure. Each node indicates a feature and splits and finally the leaf nodes indicate the class label. The main benefit of this approach is that it is resistant to non-relevant features.

Scalability of Inductive learning using Decision Tree Classifier:

Even though you add 12th to 15th attribute the accuracy of this algorithm remains the same. But the efficacy of this method is directly proportional to the valid attributes that is valid bio markers. For example, if we try adding “K” or Potassium for Decision Tree classifier the accuracy may be affected because Potassium is not a suitable marker. Only those which are valid markers if they are added then only, we will get the accuracy.

B) Deductive Learning using Rule Fit algorithm:

This algorithm merges Rule Fit and regression models. Based on feature and importance to be provided a decision tree is generated.

This algorithm is scalable because it constructs decision tree by making use of top-down approach and more suitable markers added the accuracy of this algorithm is unaffected.

Again, accuracy is only dependent on valid bio markers if irrelevant biomarkers are added the accuracy will be affected in a bad manner.

# implementation details

Software:

1) Python: For implementation of algorithms [17]

2) Excel: For storing the database.[18]

3) Anaconda Navigator: It is desktop software that allows to launch the applications. Here I made use of Anaconda Navigator to run Virtual machines.[21]

4) Visual Studio Code: Editor where I coded python applications.[22]

5) Virtual Machine in Anaconda: It is used to run the application because if we write pip install in the terminal Anaconda Navigator gets affected. [23]

6) Google Browser: To search for implementation concepts.[24]

Libraries Used:

1) NumPy (Numeric computation using Python): A popular library primarily used for handling arrays and numerical operations.[25]

2) Pandas (Python Data Analysis Library): This library is mainly used for data analysis.[26]

3) Scikit-learn: This library was used for decision tree classifier and splitting data set into training and testing (train-test-split) [27]

4) Rulefit: A library used for rule fit algorithm.[28]

5) Matplotlib: To plot the graphs. [29]

6) Graphviz: A library to create graphical visulizations

[30]

Hardware:

1) Seagate External Hard disk: It has 1 terabyte space I stored Visual Studio Code and Anaconda Navigator on this.

2) Operating System: Windows 10.

Algorithms Used

## Decision Tree Classifier for inductive learning

A picture containing diagram, text, line, plan

Description automatically generated

It is a tree like structure which mainly comprises of root node, parent node and leaf node.

The Entropy can be given as follows.

n

Entropy(S)=∑- (pi log2 pi)

i=1

pi  is probability of each class in dataset.

[14]

Information gain is given as

I(Gain)=Entropy(Parent)-[AverageEntropy(Children)]

[15]

using these formulas we can draw the decision tree.

The root of the tree being the paramount information gain and leaves are having the minimum information gain.

## Rule fit algorithm for deductive learning[19]

A picture containing text, screenshot, font, diagram

Description automatically generated

Rule fit model combines rule-based model with decision trees.

It involves five steps they are as follows.

1) Construct a decision tree: Here the decision tree is constructed based on importance.[20]

2) Extract rules from the decision tree extract the rules

3) Do a linear regression

The linear model is given as

y= B0+B1R1+B2R2…. BnRn

Where R1, R2, R3 are rules and B0, B1, …..Bn are coefficient of regression.[16]

4) Provide importance to rules

It generates a set of rules and fits a linear model by making use of these rules as attributes.

##### v. results and observations

A picture containing text, screenshot, diagram, plot

Description automatically generated

The confusion matrices for both inductive and deductive learning indicates that deductive learning is more accurate than inductive learning. For example, if accuracy of inductive learning is 80.9% the deductive learning accuracy is 84.99%.

##### vi. conculsion

This method is not as accurate as supervised learning, but it is more accurate compared to unsupervised learning and it is cost effective.

In future more focus shall be made on detecting all the stages of Ovarian Cancer.

In the contemporary world labelled data is arduous to obtain and if we make use of only unlabeled data accuracy may be affected badly.

Labelling data can be cost consuming so in this case the implementation is being made using approaches of semi supervised learning.

And if we make use of unsupervised learning it is highly inaccurate, so it is important to note that we need accuracy as well as optimal solution in determining the ovarian cancer, so the approach of semi supervised learning is being used in order to be more accurate and utilize the resources efficiently.

##### References

1. X.-X. Xi, H.-L. Wang, T. Chen, J.-R. Dai, S.-Y. Hou, and Y.-G. Chen, “Prognostic value of preoperative serum bilirubin levels in ovarian cancer,” *American journal of translational research*, vol. 12, no. 5, pp. 2267–2280, 2020, Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7269994/
2. “White blood cell count and ovarian cancer: Tests and diagnosis,” *www.medicalnewstoday.com*, Jul. 28, 2022. <https://www.medicalnewstoday.com/articles/white-blood-cell-count-ovarian-cancer>.
3. J. LAFLEUR *et al.*, “Prognostic Value of Serum Creatinine Levels in Patients with Epithelial Ovarian Cancer,” *Anticancer Research*, vol. 38, no. 9, pp. 5127–5130, Sep. 2018, doi: https://doi.org/10.21873/anticanres.12834.
4. I. S. Rasjad and B. A. Tjokroprawiro, “The Role of Procalcitonin as a Prognostic Variable in Ovarian Cancer Patients at Dr. Soetomo General Hospital Surabaya,” *Indonesian Journal of Cancer*, vol. 15, no. 3, p. 107, Oct. 2021, doi: https://doi.org/10.33371/ijoc.v15i3.718.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
8. J. Li *et al.*, “HE4 as a biomarker for ovarian and endometrial cancer management,” *Expert Review of Molecular Diagnostics*, vol. 9, no. 6, pp. 555–566, Sep. 2009, doi: <https://doi.org/10.1586/erm.09.39>.
9. S. Ye *et al.*, “Peripheral lymphocyte populations in ovarian cancer patients and correlations with clinicopathological features,” *Journal of Ovarian Research*, vol. 15, no. 1, Apr. 2022, doi: https://doi.org/10.1186/s13048-022-00977-3.
10. F. Shen, S. Chen, Y. Gao, X. Dai, and Q. Chen, “The prevalence of malignant and borderline ovarian cancer in pre- and post-menopausal Chinese women,” *Oncotarget*, vol. 8, no. 46, pp. 80589–80594, Aug. 2017, doi: <https://doi.org/10.18632/oncotarget.20384>.
11. W. Lee, S. Y. Ko, M. S. Mohamed, H. A. Kenny, E. Lengyel, and H. Naora, “Neutrophils facilitate ovarian cancer premetastatic niche formation in the omentum,” *Journal of Experimental Medicine*, vol. 216, no. 1, pp. 176–194, Dec. 2018, doi: https://doi.org/10.1084/jem.20181170.

Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou, “Learning from Incomplete and Inaccurate Supervision,” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2019, doi: <https://doi.org/10.1145/3292500.3330902>.

[13] “Intermediate ML Techniques for Detection of Cancer,” *kaggle.com*. <https://www.kaggle.com/code/tobaadesugba/intermediate-ml-techniques-for-detection-of-cancer>.

[14]

S. T, “Entropy: How Decision Trees Make Decisions,” *Medium*, Jan. 11, 2019. <https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>.

[15]

“Information Gain Which test is more informative? Split over whether applicant is employed.” Available: <https://homes.cs.washington.edu/~shapiro/EE596/notes/InfoGain.pdf>

[16]

R. Sutaria, “Linear Regression.” Available: <https://sassofia.com/wp-content/uploads/2016/05/Linear-Regression.pdf>

[17]

Python, “Welcome to Python.org,” *Python.org*, May 29, 2019. <https://www.python.org/>

[18]

Microsoft, “Microsoft Excel, Spreadsheet Software,” *www.microsoft.com*, 2022. <https://www.microsoft.com/en-us/microsoft-365/excel>

[19]

C. Molnar, “RuleFit,” *GitHub*, Apr. 25, 2023. <https://github.com/christophM/rulefit>.

[20]

G. M. K, “Machine Learning Basics: Decision Tree Regression,” *Medium*, Jul. 18, 2020. <https://towardsdatascience.com/machine-learning-basics-decision-tree-regression-1d73ea003fda>

[21]

“Anaconda Navigator — Anaconda documentation,” *docs.anaconda.com*. <https://docs.anaconda.com/free/navigator/index.html>

[22]

Wikipedia Contributors, “Visual Studio Code,” *Wikipedia*, Mar. 21, 2019. <https://en.wikipedia.org/wiki/Visual_Studio_Code>

[23]

“Virtual machine and Anaconda installation and configuration – Tutorials.” <https://fulmanski.pl/tutorials/computer-science/big-data/virtual-machine-and-anaconda-installation-and-configuration>

[24]

“Google Chrome - Download the Fast, Secure Browser from Google,” *www.google.com*. https://www.google.com/chrome/dr/download/?brand=JSBI&geo=US&gclid=Cj0KCQjwu-KiBhCsARIsAPztUF0WEHFqol8TOVYC7TINjln4yprqQR0NH9urLhL\_Utwc0BN-QwcaUQYaAqfGEALw\_wcB&gclsrc=aw.ds

‌[25]

Numpy, “NumPy,” *Numpy.org*, 2009. <https://numpy.org/>

[26]

Pandas, “Python Data Analysis Library — pandas: Python Data Analysis Library,” *Pydata.org*, 2018. https://pandas.pydata.org/

[27]

scikit learn, “sklearn.tree.DecisionTreeClassifier — scikit-learn 0.22.1 documentation,” *Scikit-learn.org*, 2019. <https://scikit->

learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[28]

Matplotlib, “Matplotlib: Python plotting — Matplotlib 3.1.1 documentation,” *Matplotlib.org*, 2012. https://matplotlib.org/

‌[29]

M. Galarnyk, “Visualizing Decision Trees with Python (Scikit-learn, Graphviz, Matplotlib),” *Medium*, Apr. 02, 2020. <https://towardsdatascience.com/visualizing-decision-trees-with-python-scikit-learn-graphviz-matplotlib-1c50b4aa68dc>

[30]

C. Gaikwad, “Build and Visualize a simple Decision Tree using Sklearn and Graphviz,” *ChiGa*, Oct. 02, 2021. https://medium.com/chinmaygaikwad/build-and-visualize-a-simple-decision-tree-using-sklearn-and-graphviz-84bda6b6b894

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌

‌