

Healthcare (Cancer, Diabetes Prediction)

A PROJECT REPORT

for

DATA MINING TECHNIQUES (ITE2006)

in

B.Tech – Information Technology and Engineering

by

Surendar SK (19BIT0388)

Parth Prakhar Mishra (19BIT0390)

Hardik Tuteja (19BIT0397)

Under the Guidance of

Dr. SENTHILKUMAR N C

Associate Professor, SITE



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

June, 2021

DECLARATION BY THE CANDIDATE

We hereby declare that the project report entitled “**Healthcare (Cancer, Diabetes Prediction)**” submitted by us to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (ITE2006)** is a record of bonafide project work carried out by us under the guidance of **Dr. Senthilkumar N C**. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course.

Place : Vellore

Signature

Date : 31st May, 2021



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled “**Healthcare (Cancer, Diabetes Prediction)**” submitted by **Parth Prakhar Mishra (19BIT0390)** to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (ITE2006)** is a record of bonafide work carried out by them under my guidance.

Dr. Senthilkumar N C

GUIDE

Associate Professor, SITE

Healthcare (Cancer, Diabetes Prediction)

Abstract

This paper presents approach for identification and prediction of one of the world's most deadliest diseases like cancer and diabetes which generally leads to several other disorders in the human body.

The prediction model has been based upon Support Vector Machine implementing major factors like robustness, accuracy and non- redundancy. The proposed method has been compared to Decision Trees and it was observed that the proposed method surpasses the efficiency by a marginal rate.

Keywords – Data Mining, Healthcare, Cancer, Diabetes, Support Vector Machine, Decision Tree Classification, PCA, Feature Scaling , Data Pre-Processing .

I. INTRODUCTION

Cancer is one of the most critical and deadliest disease since many years, also due to which approximately 10 million people die every year worldwide. The most scary part of having cancer is that it can start at any part of the body and a tumour starts growing over there, apart from it the more scary part is that around 70% deaths which are reported in India are due to late detection of cancer.

Diabetes, is considered as a simple disease which cause high blood pressure but its very lethal as compared to other disease because it causes problems to different treatments of other small diseases and the medications has to be monitored accordingly and extra care has to be taken to even control blood pressure; many people get diagnosed with diabetes at very later stage which when if detected at early stage can be controlled and won't cause much problem at later stage.

This research paper analyses several research papers which have been used for prediction of cancer and diabetes by using different input parameters. The model which will be used in the model will be Support Vector Machine based model for prediction of cancer and detection at early stage by based on input parameters and produce results with high accuracy.

II. BACKGROUND

In most of the paper which were analysed by our team, researcher mostly used Naïve Bias, K-Means, Decision Trees algorithms for prediction of cancer. But the algorithms used by them aren't much memory efficient and predicted with a lesser accuracy. Thus, we decided to use Support Vector Machine algorithm for our model which is memory efficient as compared to the previous listed algorithms and predicts with increased accuracy thus also making it time efficient. At the end of the project to compare our results we have used Decision Trees as most of the research paper we researched upon had used either Naïve Bayes or k-means algorithms for prediction of results, hence Decision Trees was chosen because of better due to lower real-time execution and accuracy.

The preliminary knowledge required for the project include the following -

- i) Basic knowledge of python: - The reader should be familiar with the concepts and certain libraries such as numpy, pandas of the python language and should have fine grip on several topics for better understanding of the concept used.
- ii) Classification: - The reader should be familiar with the concepts of classification, what it means and how are they performed on the given dataset and judging the suitable classification techniques according to the dataset.
- iii) Data Processing: - The given data is pre-processed before any operations are performed on it. Thus, knowledge about data processing is advisory.

III. Literature Survey

[1]. This paper written by, Dr. S. Senthil and B. Ayshwarya, in 2018, shows the use of AI in prediction of lung cancer using feed forward back propagation neural network. The enhanced images are trained and tested by neural network compared with sample training database. Particle Swarm Optimization (PSO) is applied to extract the features of the given input images and further process is proceed to detect the lung cancer.

[2]. This paper gives us an insight of prediction of Diabetes with various features. Their data is collected from Kaggle. Their result produced Root Mean Squared Error 0.39 and ROC area 0.88. Result of model Diabetes Prediction is shown as graphical format. This model is useful for health policy makers who can take preventive action before occurrence of diabetes in large number.

[3]. This paper briefs us on how, machine learning and image processing can be used to predict the stage of skin cancer. Data is the different images, which are enhanced first, then lesion segmentation is done on the dataset, and after feature extraction, the data is acted upon by classification algorithms.

[4]. This paper, gives us an insight of what Melanoma type of cancer is. This paper uses 3 different machine learning algorithms to draw the conclusion. The dataset used is derived from International Skin Image Collaboration (ISIC). The aim of this project is to determine the accurate prediction of skin cancer and also to classify the skin cancer as malignant or non-malignant melanoma.

[5]. In this paper, to analyze the dataset they have used Naive Bayes Tree and C4.5 decision tree-based classification technique and k-means clustering technique and evaluated the performance of each technique and found correlated feature and sub feature as a disease risk factor. They have taken risk factors like BMI, blood pressure, duration of diabetes, blood glucose level and then patient's age.

[6]. This paper shows that there is a strong correlation between diabetes and with BMI and glucose level which was derived through Apriori method. The paper has compared 3 different algorithms or methods namely Artificial Neural Network (ANN), Random Forest (RF) and K means clustering where ANN provided best accuracy of up to **75%**. They have taken into consideration data like Pregnancies Number, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function and Age.

[7]. In this paper, they have used data mining techniques such as Principle Component Analysis (PCA), k-means clustering and logistic regression algorithm and the result showed that PCA enhanced the k-means clustering algorithm and logistic regression classifier with more correctly classified data and a logistic regression accuracy of 1.98% higher because PCA application filtered out the irrelevant features. Due to this accuracy, the model is shown to be automatically predicting diabetes using the patient's health record data.

[8]. In this research paper, they have used ensemble technique with some data mining algorithms applied to a data set related to diagnosis of breast cancer using biological markers found in route blood test in order to diagnosis breast cancer and got a model as a result with Area Under Curve of **95%** and precision of **87%** through this model it is possible to create new screening tools to assist doctors. Apart from this, they also used Cross-Validation k-folds to obtain the model which also gave good results.

[9]. To predict breast cancer the above paper has used Data mining Technique such as k-nearest neighbor (KNN), Support Vector Machine (SVM) and Random forest (RF). Supervised learning classifier such as Neural network, Naive Bayes, Decision tree and CART Classifier are used to compare their performance to determine which classifier is best for Breast Cancer dataset. The final result showed SVM-RBF with accuracy **96.84%** accuracy as compared to another classifier model.

[10]. This research paper analysis shows that to conduct prediction of breast cancer models were constructed by employing one regression analysis method-Least Absolute Shrinkage and Selection Operator (LASSO), and one metaheuristic optimization method namely, Genetic Algorithm (GA). Also other data mining techniques like Random Under Sampling (RUS), Synthetic Minority Over-sampling Technique (SMOTE), Artificial Neural Network (ANN) and Logistic Regression (LR) were applied to increase the performance of the classification models.

[11]. The paper basically discovers the best highlights that are valuable for the depiction and analysis of breast cancer. K-nearest neighbour and Random Forest algorithms had been used in the study to determine the accuracy, precision, sensitivity and specificity for the cancer dataset. The accuracy calculated by this model is about **92%**. More study can be conducted on this proposed model as there is scope for improvement in effectiveness and efficiency of the model.

[12]. The paper presents an extensive review of various data mining classification techniques for prediction of cancer; the datasets contained in this study have been used in wide variety of cancer, such as breast, ovarian, etc. The most successful approach was found to be Support Vector Machine which approximately gave **99.5%** accuracy. There is a scope of improvement in the model as the number training dataset can be increased which will eventually increase precision.

[13]. The aim of the this study is choose correct data mining algorithms and increase the accuracy rate of the whole model which ultimately can help in saving millions of lives by early detection of breast cancer. The dataset purity can be increased to make the model more accurate in future studies. One of the best algorithm was found out to be k-means which gave accurate results with approximately **95%**. The model can be improved with better filtered out data in the future studies.

[14]. World has seen much advancement in the recent years but still there has been high mortality rate, therefore, timely and accurate diagnosis becomes a very integral part of the treatment. In the research has been found that if machine learning and deep learning be combined and their algorithms be used then the most of the complex cases can become simple for prediction of cancer. The ANN algorithm used provides an accuracy of **96.25%** which can be increased with proper blend with deep learning algorithms.

[15]. This paper mainly showcases how with the use of Internet of Things(IoT), Deep Learning(Branch of AI) can be used in diagnosis of Leukaemia. An IoT-enabled microscope uploads the blood smear images to the leukemia cloud. Proposed models outperform the previous approaches with average accuracy of **99.56%** and **99.91%** for ResNet-34 and DenseNet-121, respectively.

[16]. In this research paper, an optimized CNN model for classification of the type of cancer, i.e., ALL or MM, has been deployed. The classification model is built using TensorFlow, an end-to-end open-source platform. For different dataset of different cancer, this model shows accuracy figures of **82%, 87%, 97.2%**.

[17]. This paper compares 5 different algorithms to classify and predict leukemia. Firstly, it briefs the readers about the disease and then enlightens about every algorithm used. The different results are found out and compared among themselves in tabular form for better understanding. The accuracy figures are showcased as follows, SVM - **92%**, KNN – **80%**, Neural Network – **93.7%**, Naïve Bayes – **80.88%**, Deep Learning – **97.78**.

[18]. This paper compares four different classification algorithms for lung cancer prediction. Using the same data set, with the following classifiers, KNN, Naïve Bayes, Random Forest Classifier, J48 Classifier. The different results are found out and compared among themselves and it is noted that RBF gives the best result.

[19]. This paper firstly showcases various common skin cancers out there. Then, it takes reference from some previous written papers, and talks about their merits and demerits. This paper discusses various strategies for the identification and classification of skin cancer moles like convolutional neural network, transfer learning.

[20]. This paper gives a systematic review of use of AI and ML in diabetes detection. This paper compares various different classifiers and provides a thorough study of automatic diabetic detection and diagnosis techniques. The accuracy figures are, SVM – **95.83%**, Decision Trees – **90.4%**, Random Forest – **87.5%**.

[21]. This research paper shows how to process publicly available Triple Negative Breast Cancer (TNBC) gene expression datasets that is generated by Affymetrix gene chips and define a set of genes, or gene signature that can classify TNBC samples between Basal-like breast cancers (BLBCs) and Non-BLBC subtypes. Data mining methods along with dimensionality reduction and feature selection technique like Chi-square, tree, LARS, LASSO, and ensemble have been used to understand molecular characteristics and then classify them.

[22]. This research paper shows that, they have introduced a novel fuzzy methodology (IFFP Improved Fuzzy Frequent Pattern Mining), based on a fuzzy association rule mining for extraction, to analyze the dataset in order to find the core factors that causes breast cancer. This method consists of two phases. During the first phase, fuzzy frequent item sets are mined using the proposed algorithm IFFP. Fuzzy association rules are formed during the second phase, indicating whether a person belongs to benign or malignant.

[23]. This research paper analyses the pipeline of various tasks such as selecting the dataset, pre-processing the data by applying numerous methods such as standardization, normalization etc and feature extraction technique is implemented on the dataset for improving the accuracy and datasets worked on data mining and fuzzy logic various classification algorithm. The computed accuracy, in case of numerous fuzzy logic approaches, high accuracy and low complexity was found to contribute a fairly high accuracy of **96%**.

[24]. In the assessment of this paper, several machine learning algorithms has been considered for prediction of prostate cancer. The objective of the paper was to consider various models and find out the most effective one. The highest accuracy was achieved by Random Forest as compared to other algorithms which are of **90%** accuracy. Also, it was found that the Random Forest Classifiers produced best results regarding precision along with least execution time.

[25]. In this paper study was carried out by researchers to determine between different data mining algorithms which can be used for prediction of breast cancer and in classification of benign and malignant breast cancers. In the model presented by the researchers they have developed a website in which a person has to enter his/her details after which different algorithms will work upon the dataset to produce the result, thus, the algorithm which shows the best among those will be displayed on the website. Future study can be conducted on the based model to reduce the time and make the model more memory efficient.

[26]. The study was carried out to distinguish better machine learning algorithm for the prediction of malignant and benign breast cancer. Support Vector Machine algorithm was one of the algorithms which was used and produced accuracy of **99.6%**. Though, the model still has the scope of improvement which can be achieved by providing more pure datasets; also research in increasing precision and recall.

[27]. The paper is basically focused upon segmentation and classification of lung cancer by using multiple algorithms like k-nearest neighbours and Bayesian Regularisation Neural network which produces accuracy of **99.5%** and mean square error as 0.0166. The model can be improve by using a bit filtered dataset which ultimately will be increasing the efficiency of the model.

[28]. In this paper basically the SVM and KNN models comparative study has been carried out for prediction of breast cancer. The SVM model takes about 0.07s along with having an accuracy of **97.13%** to built its model as compared to KNN model which takes 0.01s with accuracy of **95.28%**. Thus SVM was found to be more powerful and accurate as compared to others. Future studies should be carried out for improvement in performance of the classification techniques to achieve more higher accuracies and reduce the error rates to a minimum.

[29]. The paper presented a comparative study of the five algorithms mentioned below, which formed basic features and working principle of each these algorithms were illustrated in detail for prediction of breast cancer. The highest accuracy obtained by ANN is **98.57%**, whereas the lowest accuracy was observed from the RFs and LR is **95.7%** . The diagnosis procedure in the medical procedure field was found to be very expensive and time consuming thus machine learning technique has been proposed.

[30]. This research paper has used data mining, machine learning algorithms (ML) and Neural Network (NN) method to predict diabetes. They have used seven ML algorithms and the model with Logistic Regression (LR) and Support Vector Machine (SVM) works well for predicting diabetes. They built NN model with different hidden layers and found that NN with two hidden layers provided 88.6% accuracy which was considered as the most efficient and promising for analysis diabetes.

IV. DATASET DESCRIPTION & SAMPLE DATA

The dataset is chosen for the diagnosis of breast cancer. There are 5 independent vectors and one dependent vector which tells that the diagnosis is required or not. The dataset is collected from *Kaggle.com*. This sample data can be used to train the model as well as some partition can be made to find the accuracy of the model.

# mean_radius	# mean_text...	# mean_peri...	# mean_area	# mean_smo...	# diagnosis
15.78	17.89	103.6	781.0	0.0971	0
19.17	24.8	132.4	1123.0	0.0974	0
15.85	23.95	103.7	782.7	0.08401	0
13.73	22.61	93.6	578.3	0.1131	0
14.54	27.54	96.73	658.8	0.1139	0
14.68	20.13	94.74	684.5	0.09867	0
16.13	20.68	108.1	798.8	0.117	0
19.81	22.15	130.0	1260.0	0.09831	0
13.54	14.36	87.46	566.3	0.09779	1
13.08	15.71	85.63	520.0	0.1075	1
9.504	12.44	60.34	273.9	0.1024	1
15.34	14.26	102.5	704.4	0.1073	0
21.16	23.04	137.2	1404.0	0.09428	0
16.65	21.38	110.0	904.6	0.1121	0

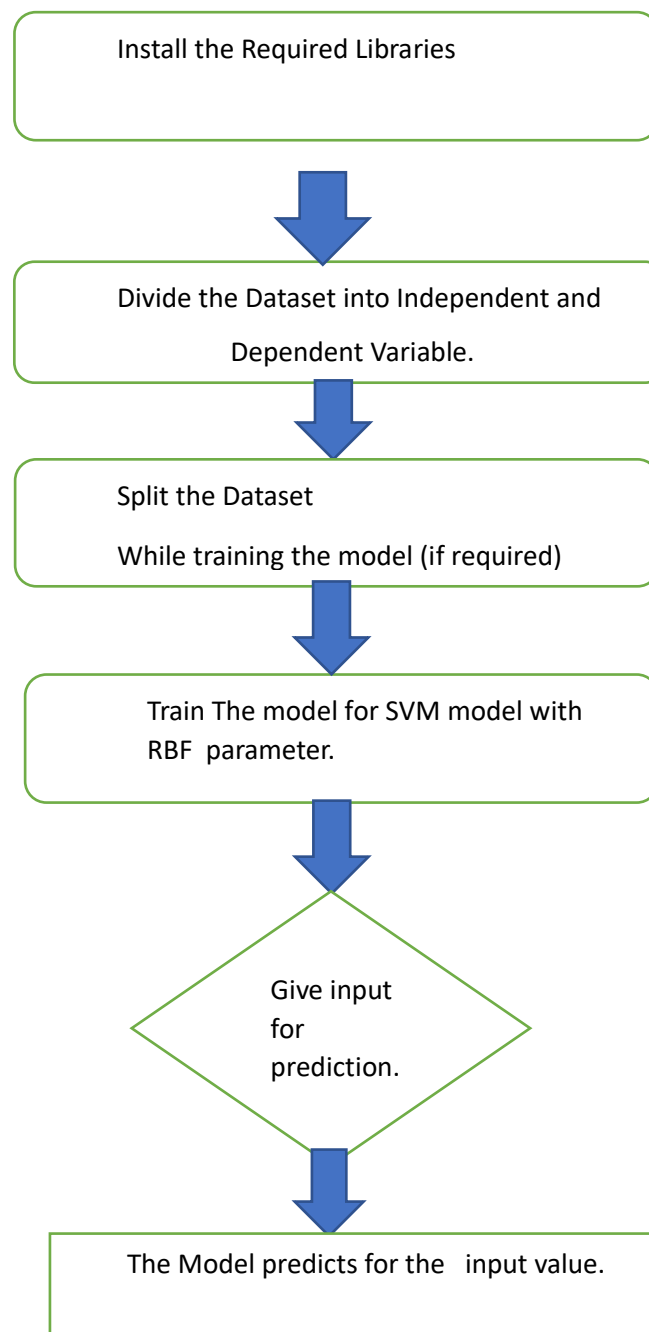
Sample Data

V. PROPOSED ALGORITHM WITH FLOWCHART

Architecture with Flowchart

After going through many papers, we have concluded that for predictions in medicinal industry, best classification technique would be SVM with 'RBF' kernel. It is a supervised learning method. It is chosen because of its extra-ordinary working mechanism which takes extra care of the border-line data points, and thus delivering better results in most of the cases.

The flowchart for the algorithm is: -



Part 1Installing all dependencies and required files

Part 1 – Installing all dependencies and required files: -

1.1 Pip install package_name can be used to download for the given packages. The packages that are needed to install before start to code are :

nlTK — for performing natural language processing tasks and model training.

lxml — it is the package that is used to process Html and XML with python.

urllib — for requesting a webpage.

sklearn(optional) — Used for saving the trained model.

Part 2Extracting data of particular cancer/ diabetes

Part 2 - Extracting data of particular cancer/ diabetes :-

2.1 – Extracting all the data from the .csv file by the help of specific packages and functionalities.



Part 3



Model Development (Support Vector Machine)

Part 3 – Model Development (Support Vector Machine): -

3.1 – Download all the necessary data like diabetes_data on which the model has to be trained.

3.2 – Using Support Vector Machine all data is stored

3.3 – The prediction phase follows next



Part 4



Prediction of Cancer/ Diabetes

Part 4 – Prediction of Cancer/ Diabetes :-

4.1 – After extraction of all the data required for evaluation, prediction is done.

4.2 – Review is given by the model whether the person have a particular disease or not.

4.3 – If percentage is higher than 90%, then it predicts that the person has that particular disease.

Proposed Algorithm -

In this project, we are using Support Vector Machine (SVM) as our data mining functionality.

A Support Vector Machine (SVM) is a binary linear as well as non-linear with 'rbf' kernel classification whose decision boundary is explicitly constructed to minimize generalization error. It is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression and even outlier detection.

It consists of four parts -

- 1) Generate Hyperplanes.
- 2) Distance Measure.
- 3) Classify Separation.
- 4) Evaluating Accuracy.

Step 1: Generate Hyperplanes -

Here, we use datasets that can be classified as either of one class among two. Each data points have a pair of co-ordinates (x, y). Using these co-ordinates, we plot the data on a plane. These points are separated using a line and this line is called as a hyperplane or decision boundary, which classifies them as separate classes. This is done to differentiate them into two non-overlapping classes.

Step 2: Distance Measure -

After establishing the hyperplane, we want an optimal hyperplane that would be able to separate the data points or data sets in the best possible way having the least mistakes or errors of miss-classification. So to have least errors in the classification of the data points, we first need to find the distance between a data point and the hyperplane. In this case, the data sets are considered as a vector. So the line equation of a particular vector would be $w^T\Phi(x) + b = 0$ and the distance between the line and the vector is calculated using the formula:

$$d_H(\phi(x_0)) = \frac{|w^T(\phi(x_0)) + b|}{||w||_2} \quad ||w||_2 =: \sqrt{w_1^2 + w_2^2 + w_3^2 + \dots w_n^2}$$

Step 3: Classify Separation -

If the data sets are differentiated into classes without any miss-calculator, then that type of separation is called Perfect Separation. In this type of separation, the hyperplane gives us 100% accuracy. The optimal hyperplane can be chosen by placing the hyperplane at the center where the distance is maximum from the closest point or the margin should be maximum from the data points. Margins are spaces around the hyperlane. So rescaling of the distance of the hyperplane is done by the formula,

$$w^* = \arg_w \max \frac{1}{||w||_2} \quad , \quad s. t. \quad \min_n y_n [w^T \phi(x_n) + b] = 1$$

$$y_n [w^T \phi(x) + b] = \begin{cases} \geq 0 & \text{if correct} \\ < 0 & \text{if incorrect} \end{cases}$$

Where $\arg \max$ is arguments of the maxima which are basically the data points of the domain of a function at which function values are maximized.

If there are data points that are not separated or there are miss-classifications, then that type of separation is called, Non-Perfect Separation. In order to correct those miss-classifications, we add a variable as a penalty for every miss-classification for each data point represented by β (beta). So, no penalty means the data point is correctly classified, $\beta = 0$, and at any miss classification $\beta > 1$, as a penalty. So rescaling of the hyperplane is done by,

Instead of $y_n [w^T \phi(x_n) + b] > 0 ; \forall n$

Equation becomes $y_n [w^T \phi(x_n) + b] \leq 0, \exists n$

Step 4: Evaluating Accuracy -

Here kernels are used to transform an input data space into the required form. SVM uses a technique called the kernel trick. Here, the kernel takes a low-dimensional input space and transforms it into a higher dimensional space. In other words, we can say that it converts Non-Perfect Separation problem to Perfect separable problems by adding more dimension to it. Kernel trick helps us to build a more accurate classifier. Using this classifier, we will be able to find the accuracy of the model.

VI. EXPERIMENTS RESULTS

The Kernel SVM classification model used for the prediction of Cancer/ diabetes gives some outstanding results. For the datasets used the accuracy in the corresponding models were as follows: -

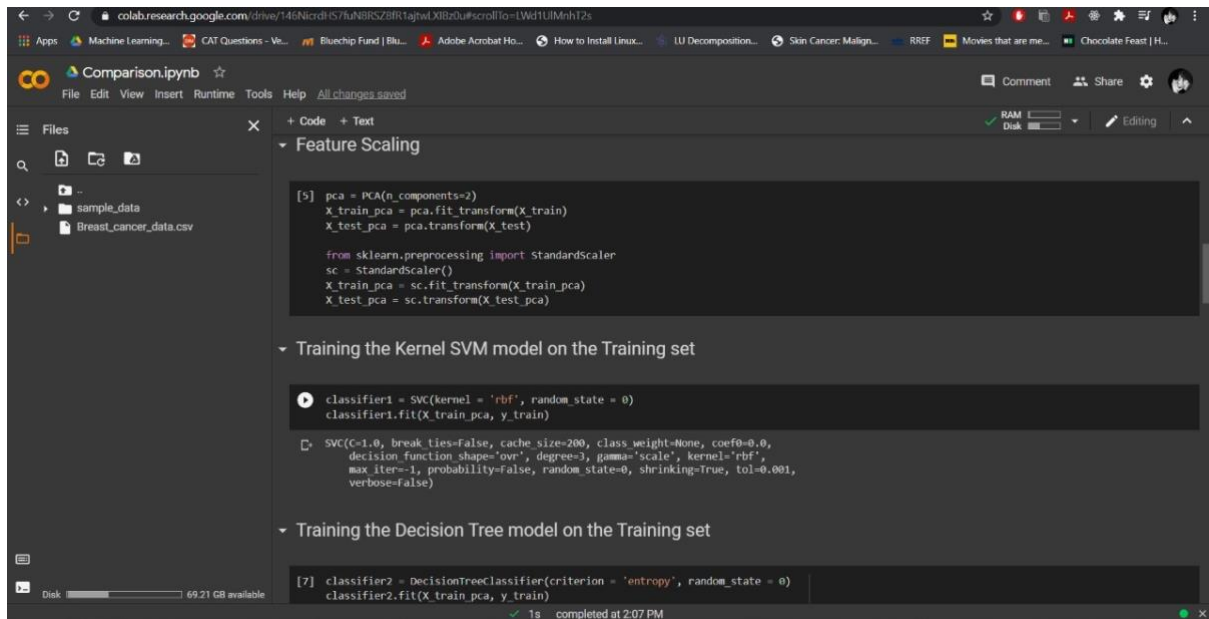
- 1) Breast Cancer: - 90.9%
- 2) Diabetes – 81.81%
- 3) Lung Cancer – 100%

These results most certainly show some promising effect and can be relied upon.

VII. COMPARATIVE STUDY / RESULTS AND DISCUSSION

The chosen methodology used kernel SVM as the classification model and when compared with another classification model called, Decision Tree, the results were as follows.

After fitting both the models on the same dataset,



```
[5] pca = PCA(n_components=2)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_pca = sc.fit_transform(X_train_pca)
X_test_pca = sc.transform(X_test_pca)

> Training the Kernel SVM model on the Training set

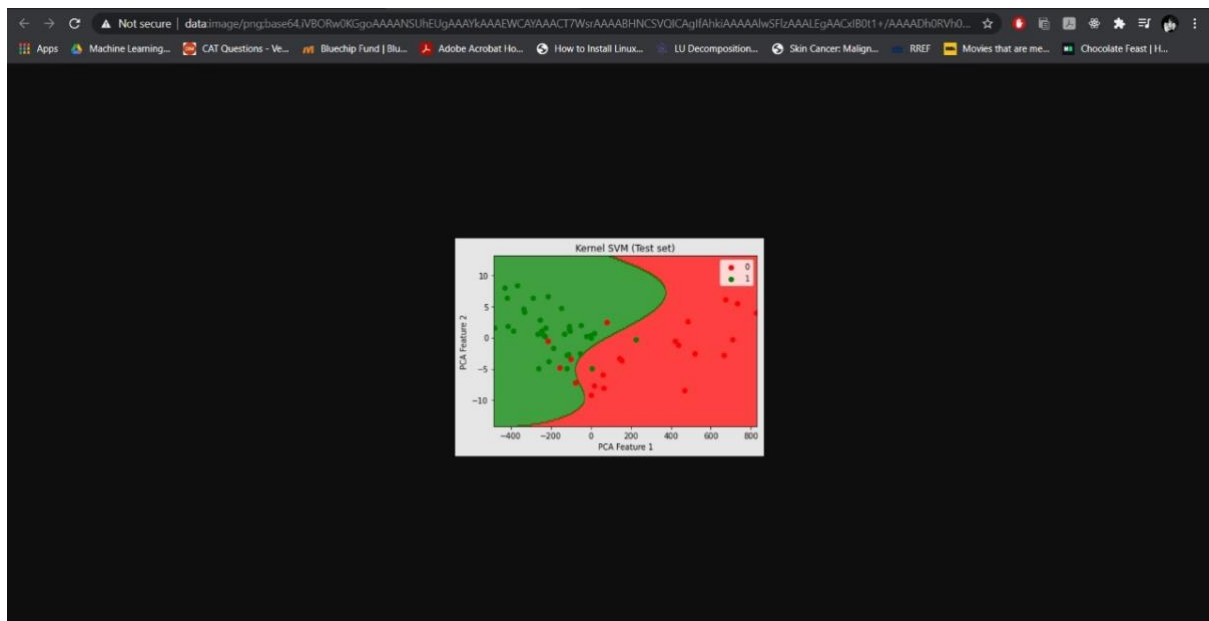
classifier1 = SVC(kernel = 'rbf', random_state = 0)
classifier1.fit(X_train_pca, y_train)

> Training the Decision Tree model on the Training set

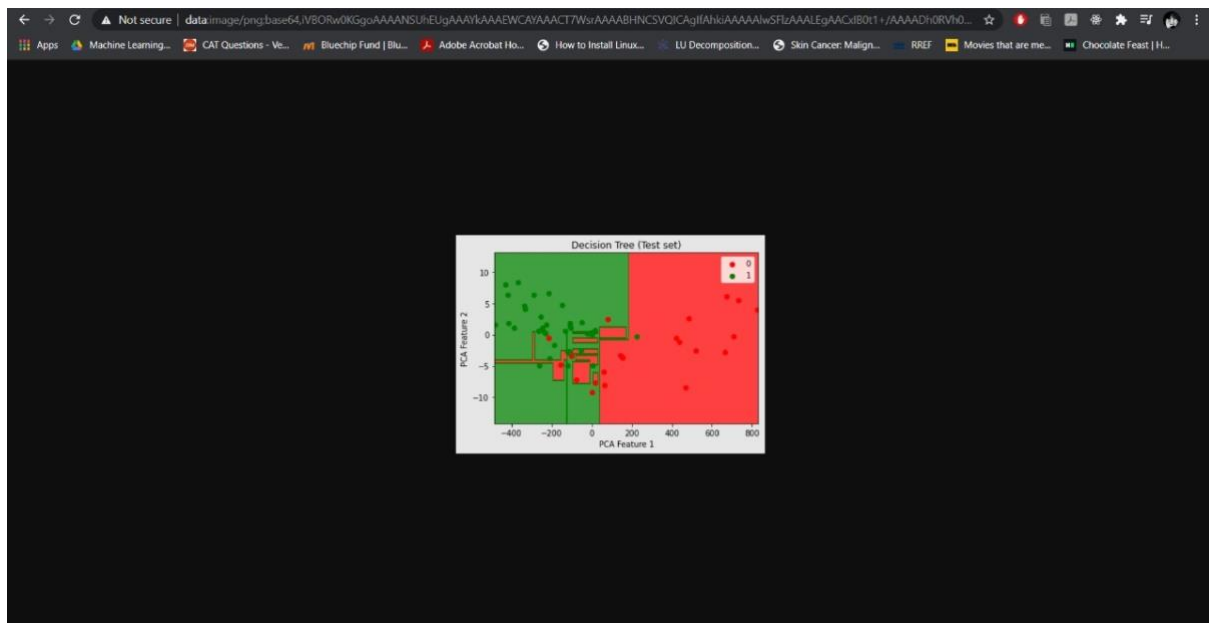
[7] classifier2 = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier2.fit(X_train_pca, y_train)
```

The visual prediction on the test set for the models were: -

SVM: -



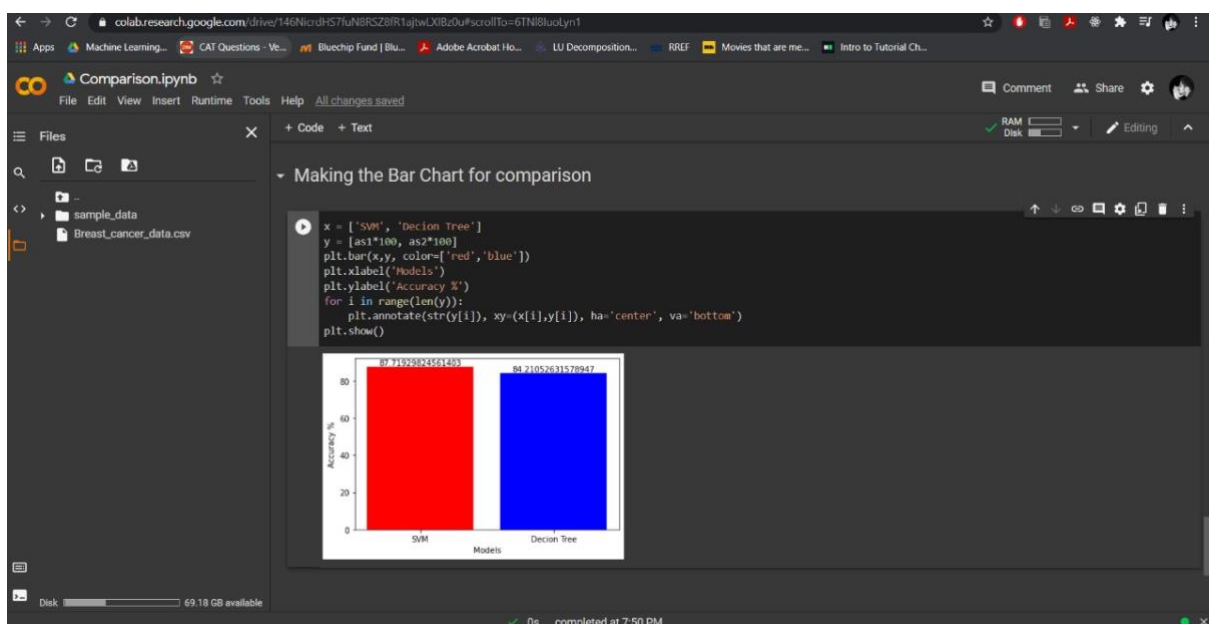
Decision Tree: -



The accuracy of the corresponding models came out to be: -

SVM – 87.71%

Decision Tree – 84.21%



VIII. CONCLUSION AND FUTURE WORK

The comparison between the results predicted by Support Vector Machine and Decision Tree algorithms show that Support Vector Machine is slightly better and much more memory efficient thus making it time efficient algorithm due to being a simpler algorithm as compared to other various algorithms discussed earlier.

Though this algorithm (Support Vector Machine) used in this research model does provide good results in prediction of cancer and diabetes but there is always scope for improvement, therefore, our model can also be improved by replacement with much advanced algorithms like Recurrent Neural Networks, Artificial Neural Networks, Convolutional Neural Networks and by using different metrics in different datasets for the prediction model. With professional help this model can be used in hospitals/ clinics to produce much more accurate results in prediction of cancer which can ultimately save lives.

IX. REFERENCES

1. Dr. S. Senthil and B. Ayshwarya, *Lung Cancer Prediction using Feed Forward Back Propagation Neural Networks with Optimal Features*, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1, 2018.
2. Lokesh Sharma, Vijeta Sharma, Ajai Kumar, Hemant Darbari, Suyash Srivastava, *Prediction of Diabetes Using Artificial Neural Network Approach: ICoEVCI*, 2018.
3. Saranya N, Nirmala M, Kiruthika P, *MACHINE LEARNING ALGORITHM FOR AUTOMATED SKIN CANCER PREDICTION AND PROGNOSIS*, INTERNATIONAL JOURNAL OF CURRENT ENGINEERING AND SCIENTIFIC RESEARCH (IJCESR), VOLUME-5, ISSUE-4, 2018.
4. Vijayalakshmi M M, *Melanoma Skin Cancer Detection using Image Processing and Machine Learning*, International Journal of Trend in Scientific Research and Development (IJTSRD), Volume: 3, Issue: 4 2019.
5. Cut Fiarni, Evasaria M. Sipayung, Siti Maemanuh, *Analysis and prediction of diabetes Complication Disease using Data Mining Algorithm*, Procedia Computer Science, Vol 161, 2019.
6. Talha Mahboob Alam, Muhammad Atif Iqbal, Yasir Ali, Abdul Wahab, Safdar Ijaz, Talha Imtiaz Baig, Ayaz Hussain, Muhammad Awais Malik, Muhammad Mehdi Raza, Salman Ibrar, Zunish Abbas, *A model for early prediction of diabetes*, Informatics in Medicine Unlocked Volume 16, 2019.
7. Changsheg Zhu, Christian Uwa Idemudia, Wenfang Feng, *Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques*, zhuInformatics in Medicine Unlocked, Vol 17, 2019.

8. Maria Ines Cruz, Jorge Bernardino, *Data mining techniques for Early Detection of Breast Cancer*, Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, doi: 10.5220/0008346504340441, 2019.
9. Elsevier B.V, *Application of Data Mining Techniques to Predict Breast Cancer*, Procedia Computer Science, Vol 163, 2019.
10. Serhat Simsek, Ugur Kursuncu, Eyyub Kibis, Musheera AnisAbdellatif, Ali Dag, *A hybrid data mining approach for identifying the temporal effects of variables associated with breast cancer survival*, Expert Systems with Applications, Vol 139, 2019.
11. Dr. B. Santhosh Kumar, T. Daniya, Dr. J.Ajayan, *Breast Cancer Prediction Using Machine Learning Algorithms*, International Journal of Advanced Science and Technology Vol. 29, No. 03, (2020), pp. 7819 – 7828, February 2019.
12. Ajay Kumar, Rama Sushil and Arvind Kumar Tiwari, *Machine Learning based Approaches for Cancer Prediction: A Survey*, 2nd International Conference on Advanced Computing and Software Engineering, April 2019.
13. R. Preetha, S. Vinila Jinny, *A Research on Breast Cancer Prediction using Data Mining Techniques*, International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol 8, Issue 11S2, September 2019.
14. Fuat Turk, Murat Luy, Necaattin Bariser, *Machine Learning of Kidney Tumors and Diagnosis and Classification by Deep Learning Methods*, International Journal of Engineering Research and Development, Vol 11, Issue 3, December 2019.
15. Nighat Bibi, Misba Sikandar, Ikram Ud Din, Ahmad Almogren, and Sikandar Ali, *IoMT-Based Automated Detection and Classification of Leukemia Using Deep Learning*, Journal of Healthcare Engineering, Volume 2020, Article ID 6648574, 2020.

16. Deepika Kumar, Nikita Jain, Aayush Khurana, Sweta Mittal, Suresh Chandra Satapathy, Roman Senkerik and Jude D. Hemanth, *Automatic Detection of White Blood Cancer From Bone Marrow Microscopic Images Using Convolutional Neural Networks*, IEEE Access, Volume 8, 2020.
17. Italia Joseph Maria, T. Devi, D. Ravi, *Machine Learning Algorithms For Diagnosis Of Leukemia*, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, ISSUE 01, 2020.
18. Radhanath Patra, *Prediction of Lung Cancer Using Machine Learning Classifier*, Springer Nature Singapore Pte Ltd., CCIS 1235, pp. 132–142, 2020.
19. Prof. Shashank Bholane, Shubham Patil, Gaurav Rajput, Swapnil Patil, Sanket Gunjalkar, *Skin Cancer Prediction using Image Processing and Deep Learning*, International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 02, 2020.
20. Jyotismita Chaki, S. Thillai Ganesh, S.K Cidham, S. Ananda Theertan (of our university), *Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management*, Article in Press, 2020.
21. Amir Hassan Zadeh, Qamar Alsabi, Jaime E. Ramirez-Vick, Nasim Nosoudi, *Characterizing basal-like triple negative breast cancer using gene expression analysis: A data mining approach*, Expert Systems with Applications, Vol 148, 2020.
22. F. Ramesh Dhanaseelan, M. Jeya Sutha, *Detection of Breast Cancer Based on Fuzzy Frequent Itemsets Mining*, IRBM Journal, 2020.
23. Harshil Thakkar, Vaishnavi Shah, Hiteshri Yagnik, Manan Shah, *Comparative anatomization of data mining and fuzzy logic techniques used in diabetes prognosis*, Clinical eHealth, Vol 4, 2020.

24. Muktevi Srivenkatesh, *Prediction of Prostate Cancer using Machine Learning Algorithms*, International Journal of Recent Technology and Engineering (IJRTE), Vol-8, Issue 5, January 2020.
25. Nikita Rane, Jean Sunny, Rucha Kanade and Prof. Sulochna Devi, *Breast Cancer Classification and Prediction using Machine Learning*, International Journal of Engineering Research & Technology (IJERT), Vol 2, Issue 2, February 2020.
26. Borislava Petrova Vrigazova, *Detection of Malignant and Benign Breast Cancer Using the ANOVA-BOOTSTRAP-SVM*, Journal of Data and Information Science, Vol. 5, No. 2, April 2020.
27. Prasanta Das, Biplab Das and Himadri Sekhar Dutta, *Prediction of Lung Cancer using machine learning*, Kalyani Government Engineering College Bulletin, April 2020.
28. Ramik Rawal, *breast cancer prediction using machine learning*, JETIR May 2020, Volume 7, Issue 5, May 2020.
29. Md. Milon Islam, Md. Rezwanul Haque, Md. Munirul Hasan, Mahmudul Hasan and Muhammad Nomani Kabir, *A Comparative Study Using Machine Learning Techniques*, SN COMPUT. SCI. Vol 1, No. 290, September 2020.
30. Jobeda Jamal Khanam Simon Y. Foo, *A comparison of machine learning algorithms for diabetes prediction*, ICT Express bulletin, February 2021.

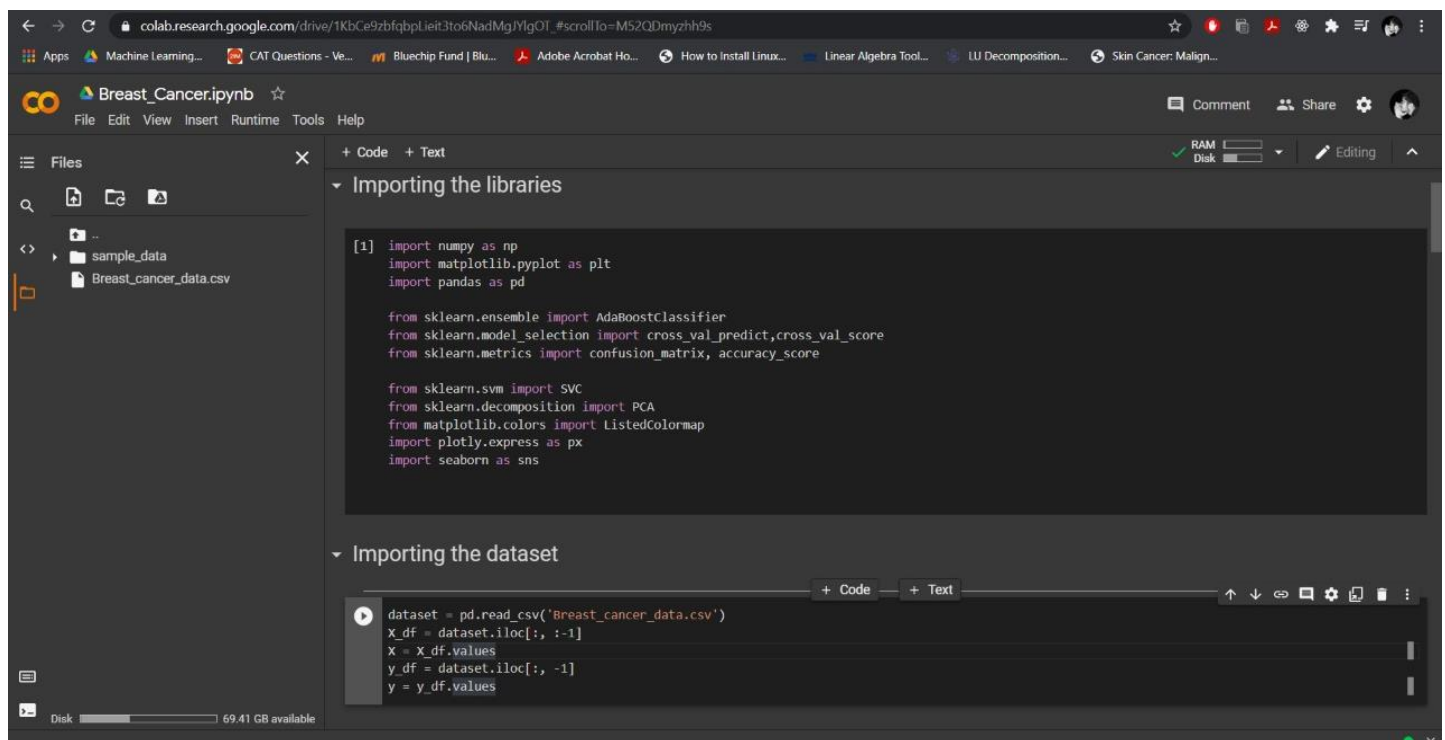
Appendix

Code Screenshots –

Breast Cancer Prediction –

Importation of Libraries –

In this screenshot, two cells of the code are shown, the first cell import all the libraries required for the project. The second cell however is used to read the test, and furthermore segregate the dataset in independent variables and dependent variable.



```
[1] import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

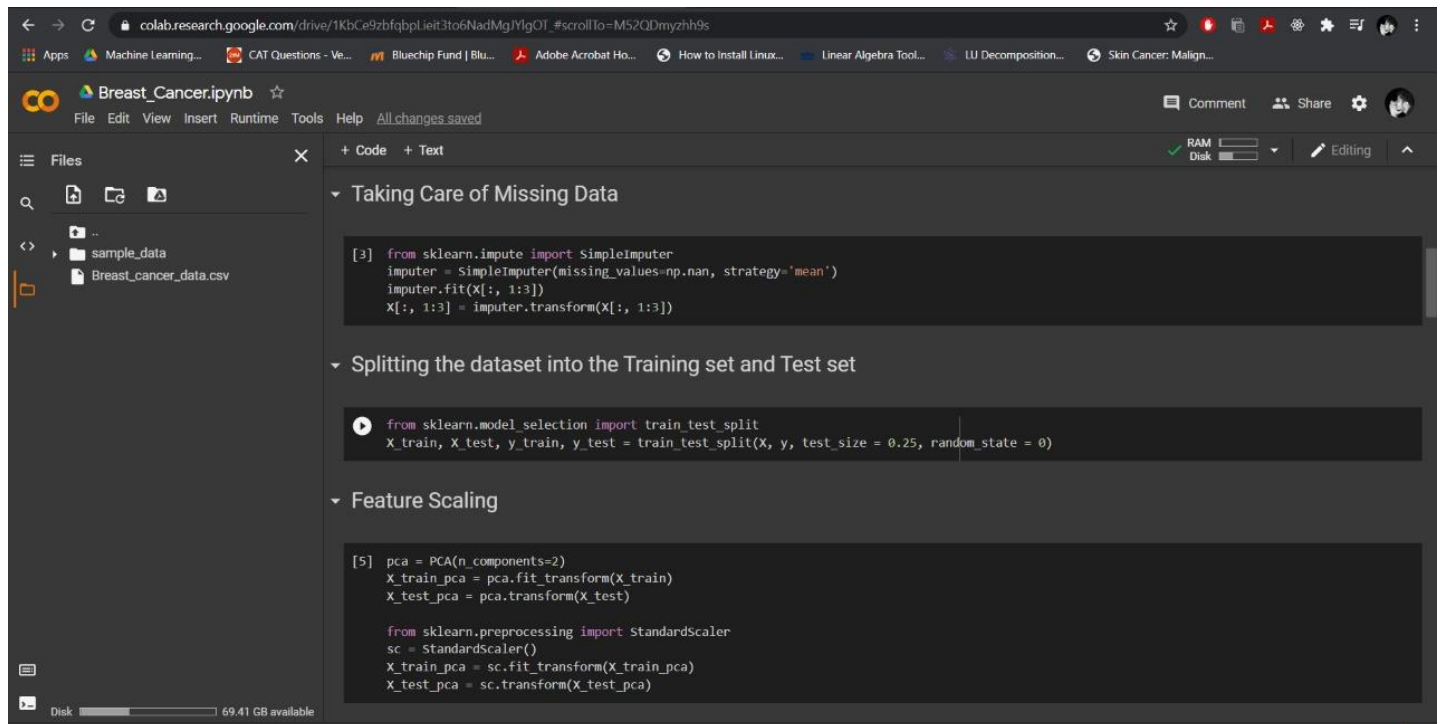
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_predict, cross_val_score
from sklearn.metrics import confusion_matrix, accuracy_score

from sklearn.svm import SVC
from sklearn.decomposition import PCA
from matplotlib.colors import ListedColormap
import plotly.express as px
import seaborn as sns
```

```
dataset = pd.read_csv('Breast_cancer_data.csv')
X_df = dataset.iloc[:, :-1]
X = X_df.values
y_df = dataset.iloc[:, -1]
y = y_df.values
```

Missing Data –

All the cells in this screenshot are parts of data pre-processing, as mentioned, the first cell specifically replaces the missing data of a particular column by the mean value of that particular column. The second cell splits the dataset into training test and testing set. The third cell comprises of two things, it is being used to reduce the dimensions of the given dataset to have a visual graphic second part performs standard scaling.



Visualising Result on Training Set –

The first cell in the code is the demonstration of the model being fitted on the dataset. This particular piece of code uses matplotlib library to plot the classification. The two axes represent the result obtained from dimensional reduction from pca code. The green area shows the positive result (1) and the red area shows the negative (0) result. This first plot is made off the training set and the second piece of code does the same for the test set and plot shows the result predicted by the machine.

colab.research.google.com/drive/1KbCe9zbfbpLiet3to6NadMg/YlgOT_

Breast_Cancer.ipynb

File Edit View Insert Runtime Tools Help Last edited on April 14

Table of contents

- Kernel SVM
 - Importing the libraries
 - Importing the dataset
 - Taking Care of Missing Data
 - Splitting the dataset into the Training set and Test set
 - Feature Scaling
 - Training the Kernel SVM model on the Training set
 - Visualising the result on training set
 - Making the Confusion Matrix

Section

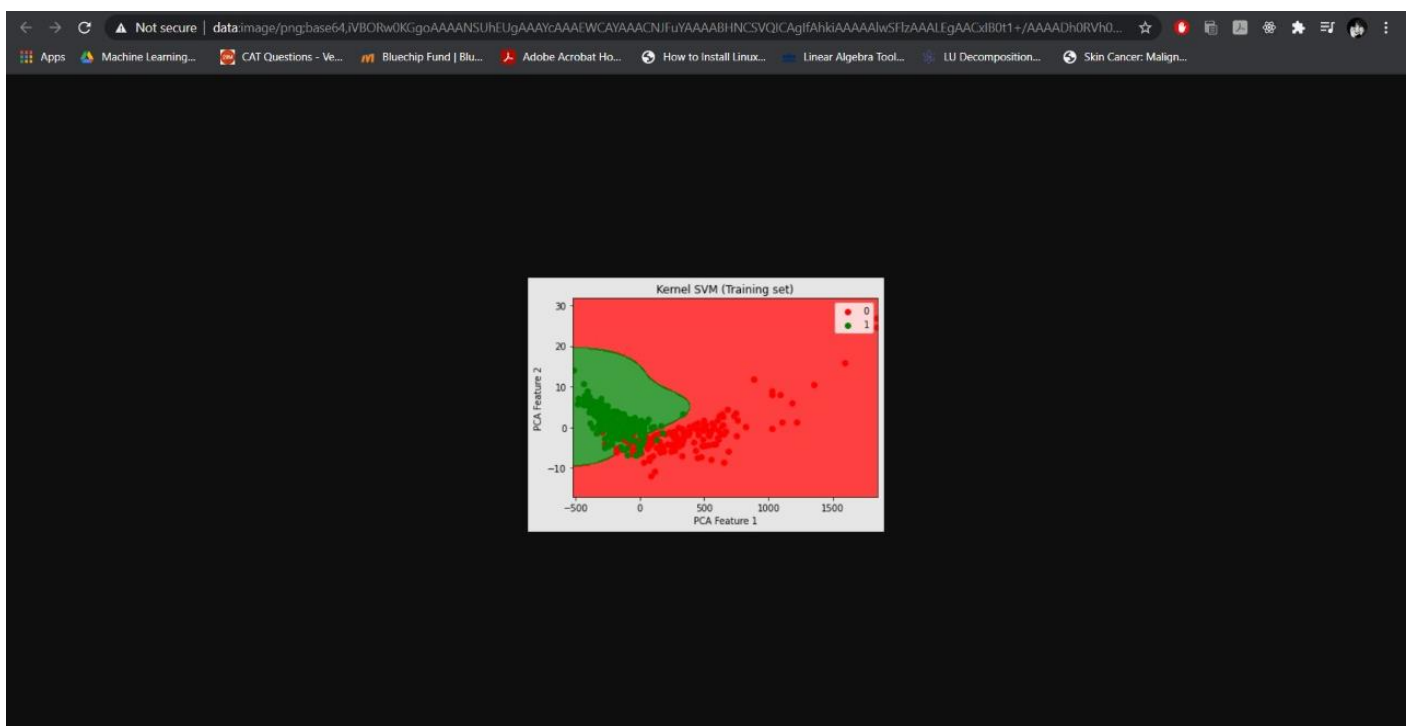
Training the Kernel SVM model on the Training set

```
[ ] classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train_pca, y_train)

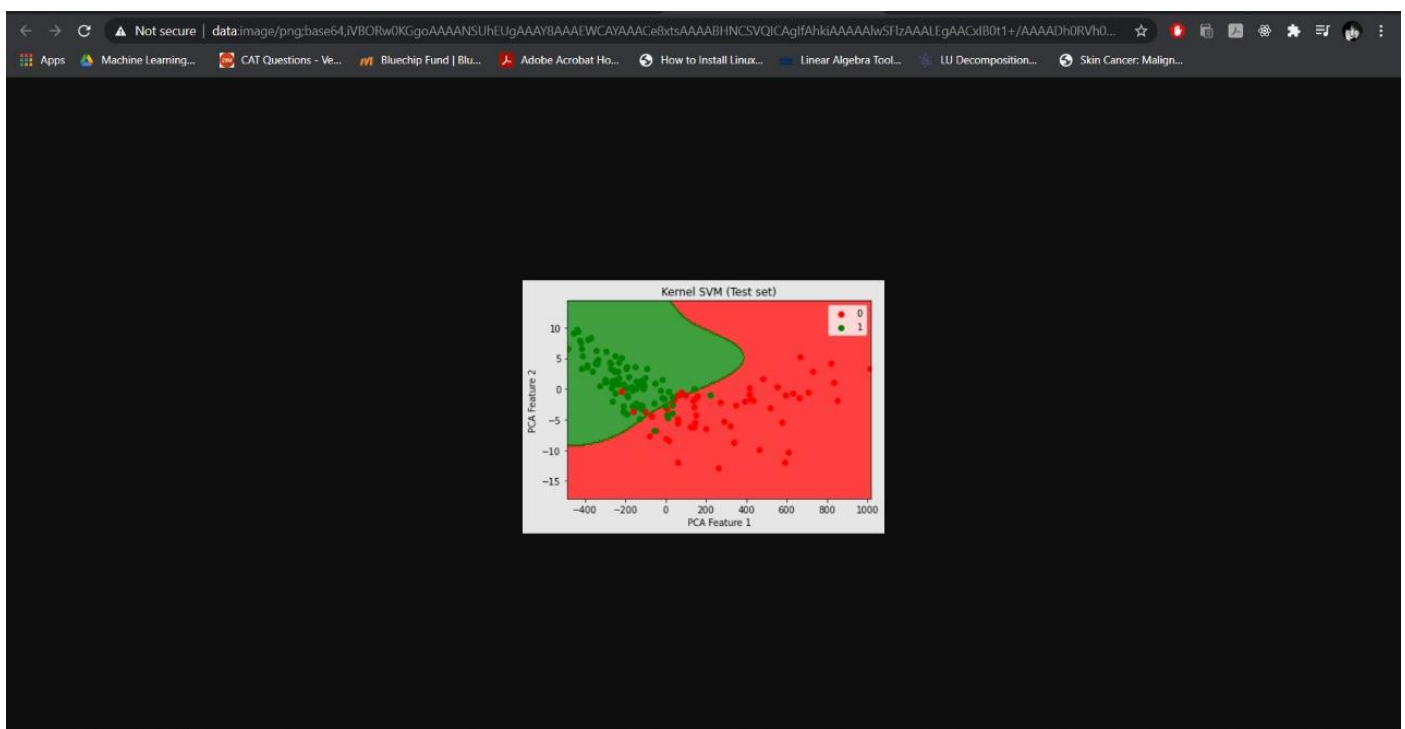
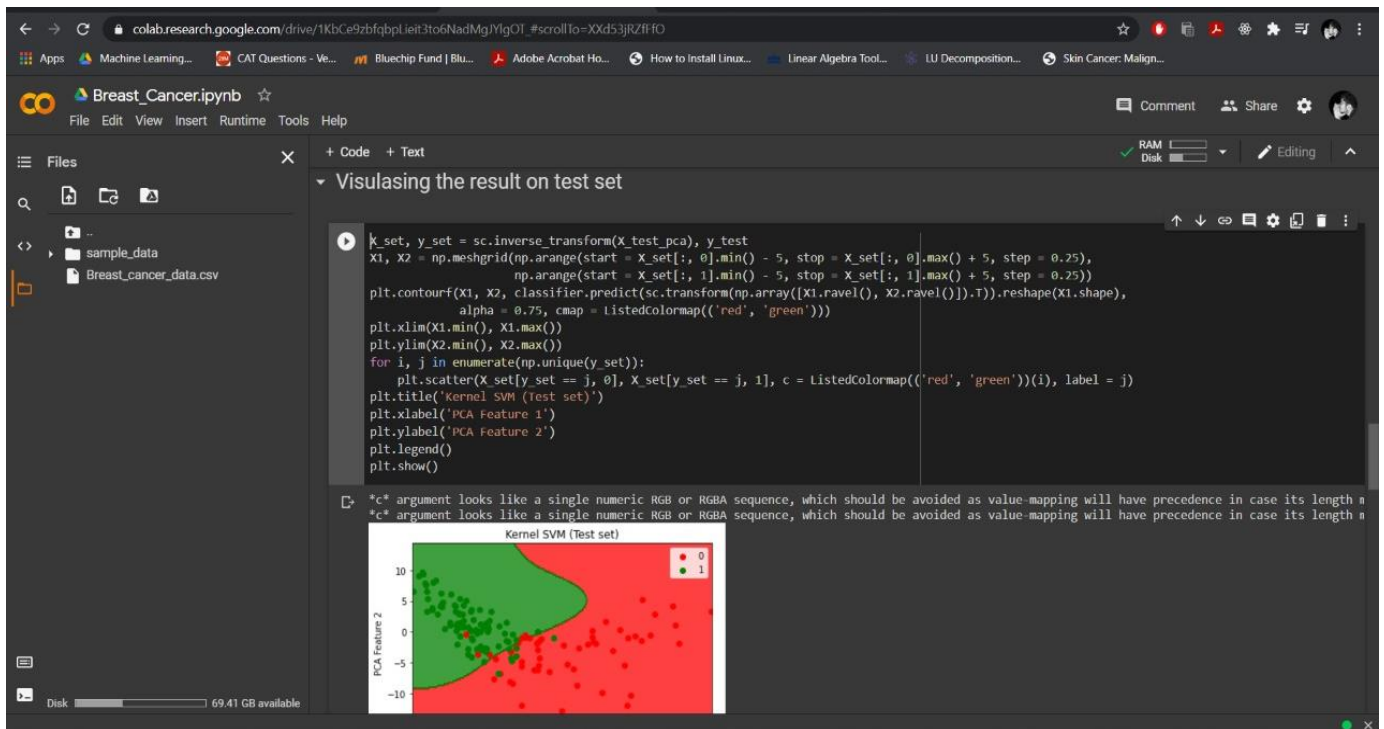
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

Visualising the result on training set

```
X_set, y_set = sc.inverse_transform(X_train_pca, y_train)
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 5, stop = X_set[:, 0].max() + 5, step = 0.25),
    np.arange(start = X_set[:, 1].min() - 5, stop = X_set[:, 1].max() + 5, step = 0.25))
res = classifier.predict(sc.transform(np.array([X1.ravel(), X2.ravel()]).T)).reshape(X1.shape)
plt.contourf(X1, X2, res,
    alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Kernel SVM (Training set)')
plt.xlabel('PCA Feature 1')
plt.ylabel('PCA Feature 2')
plt.legend()
plt.show()
```

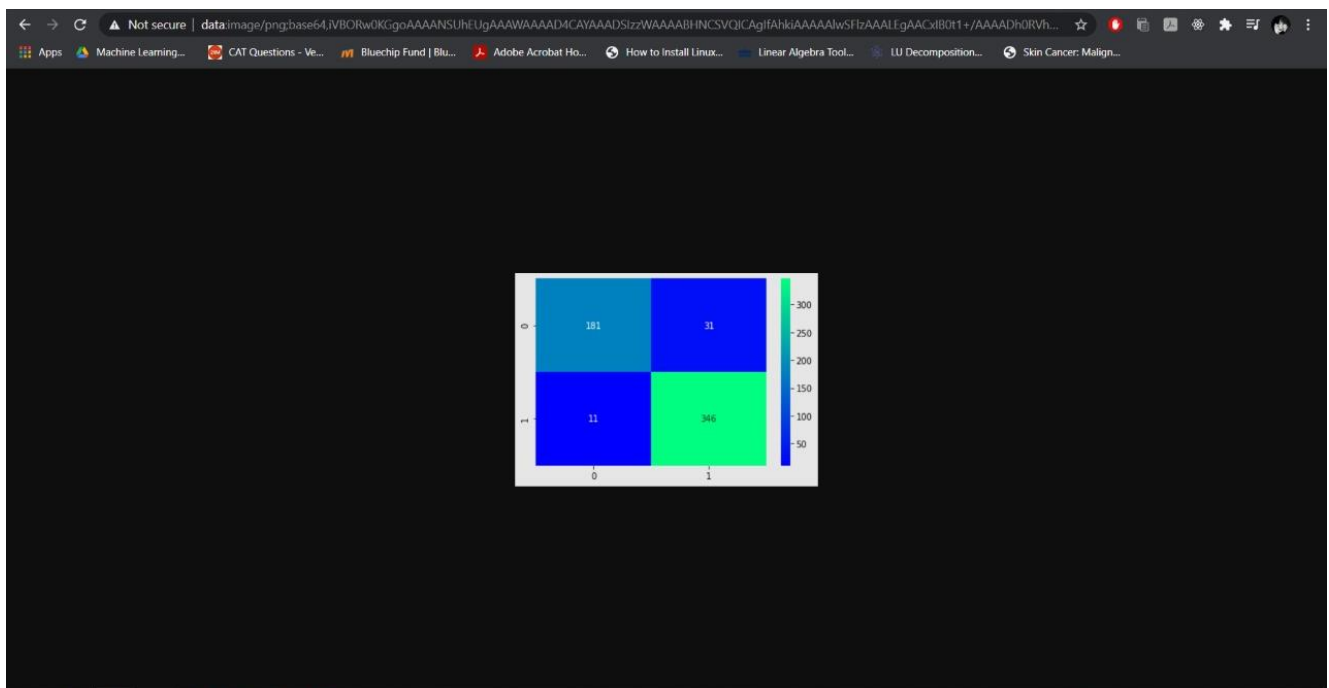
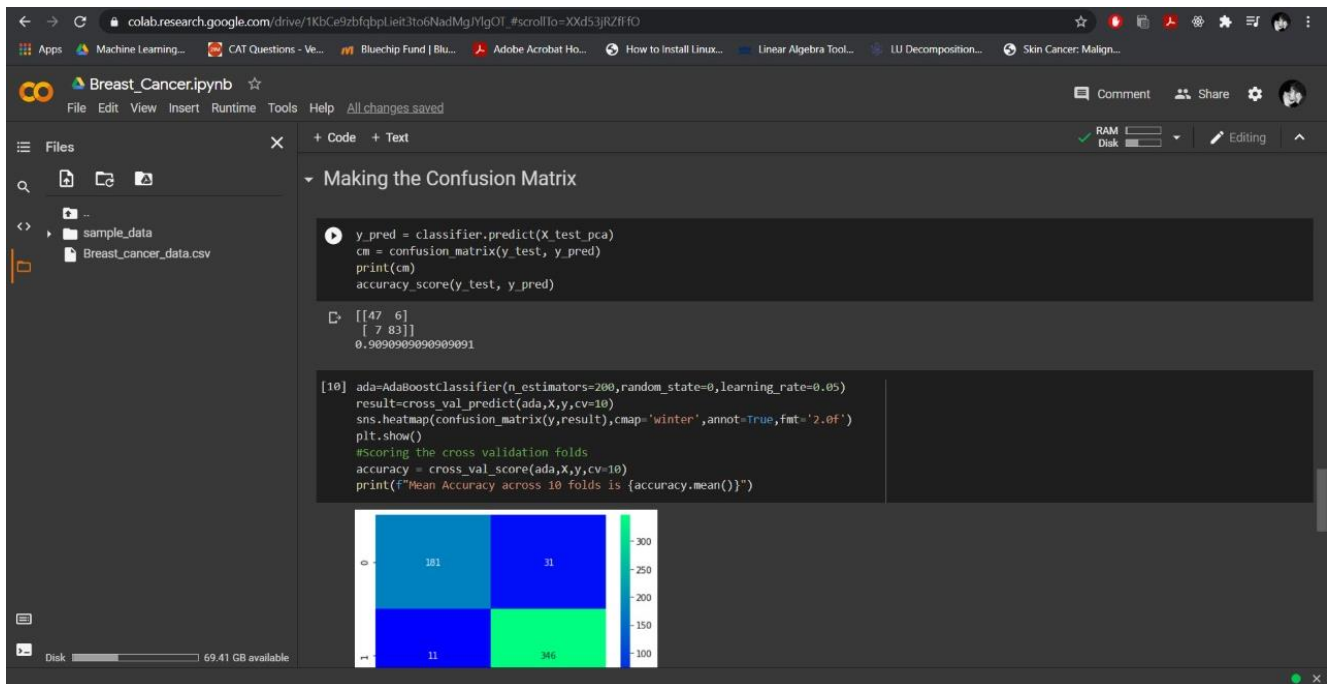


Visualising Result on Test Set –



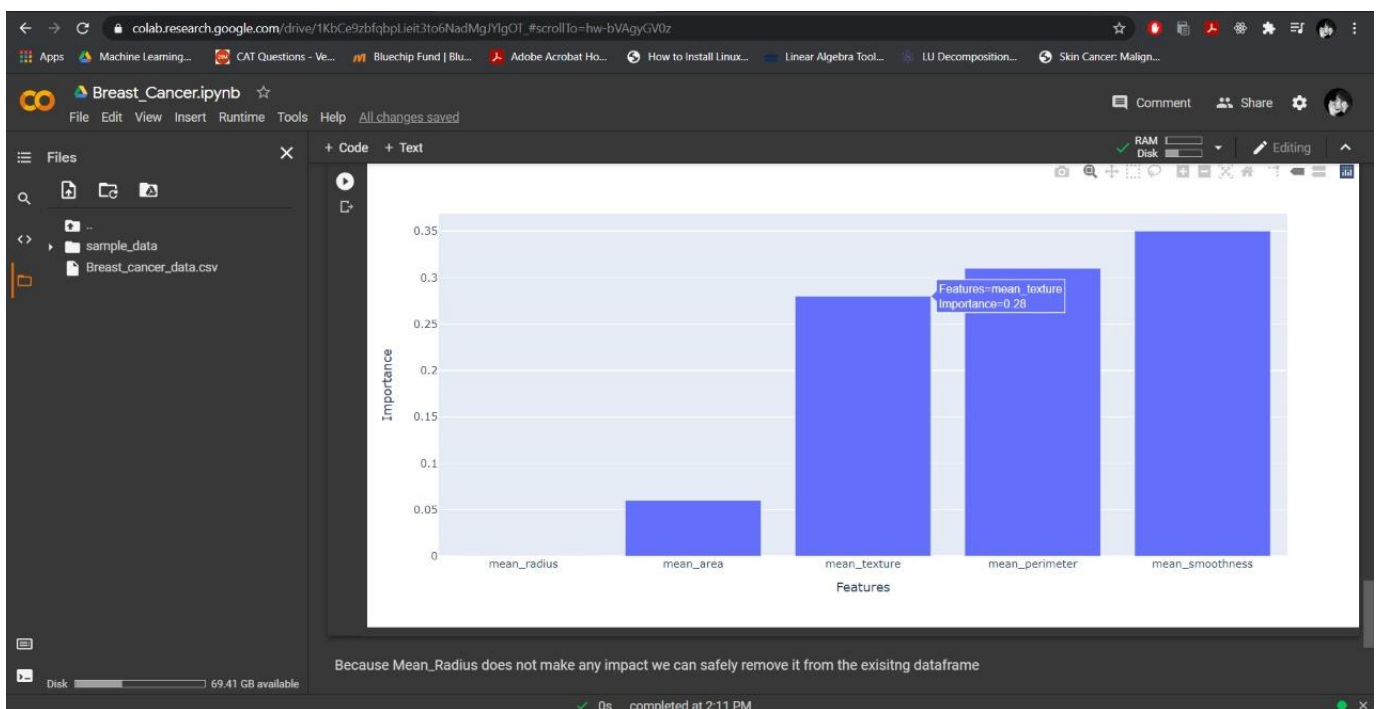
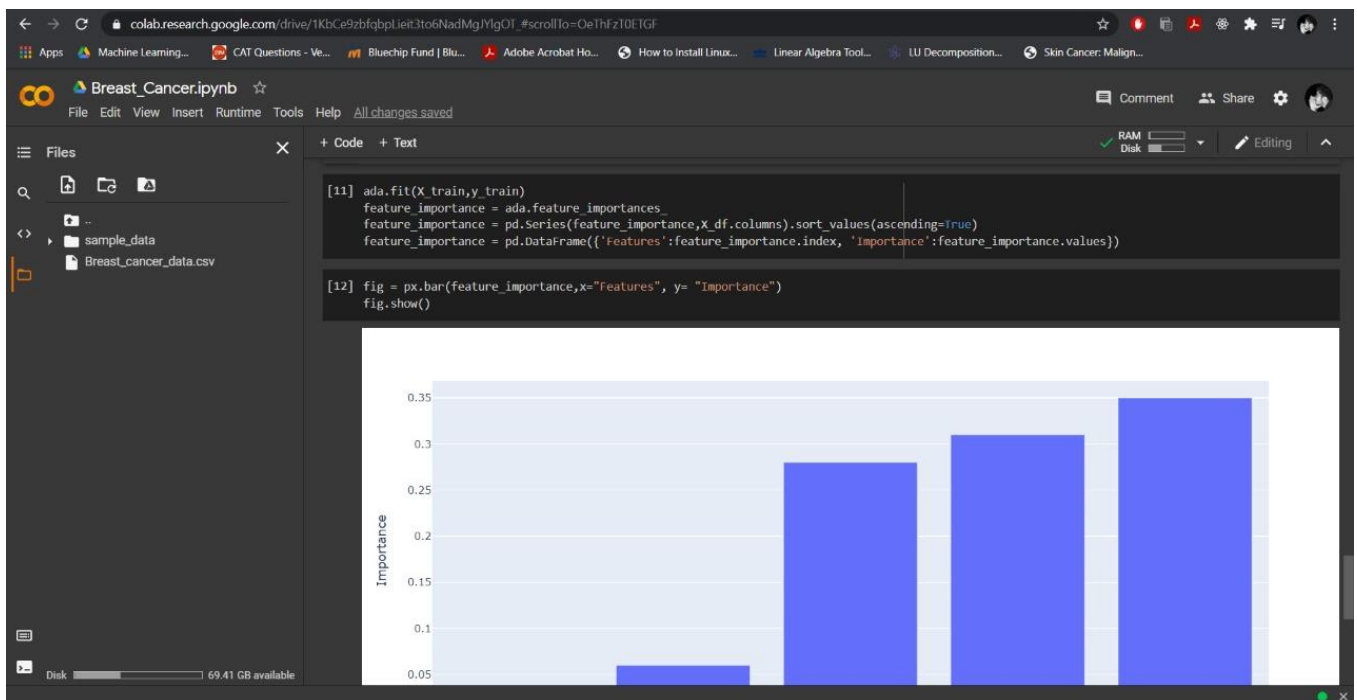
Making of Confusion Matrix –

This piece of code makes the confusion matrix and gives us the accuracy of the result which shows a promising 90.09 percent and there is interactive confusion matrix of all the learnings using adaboost.



Graphical Representation –

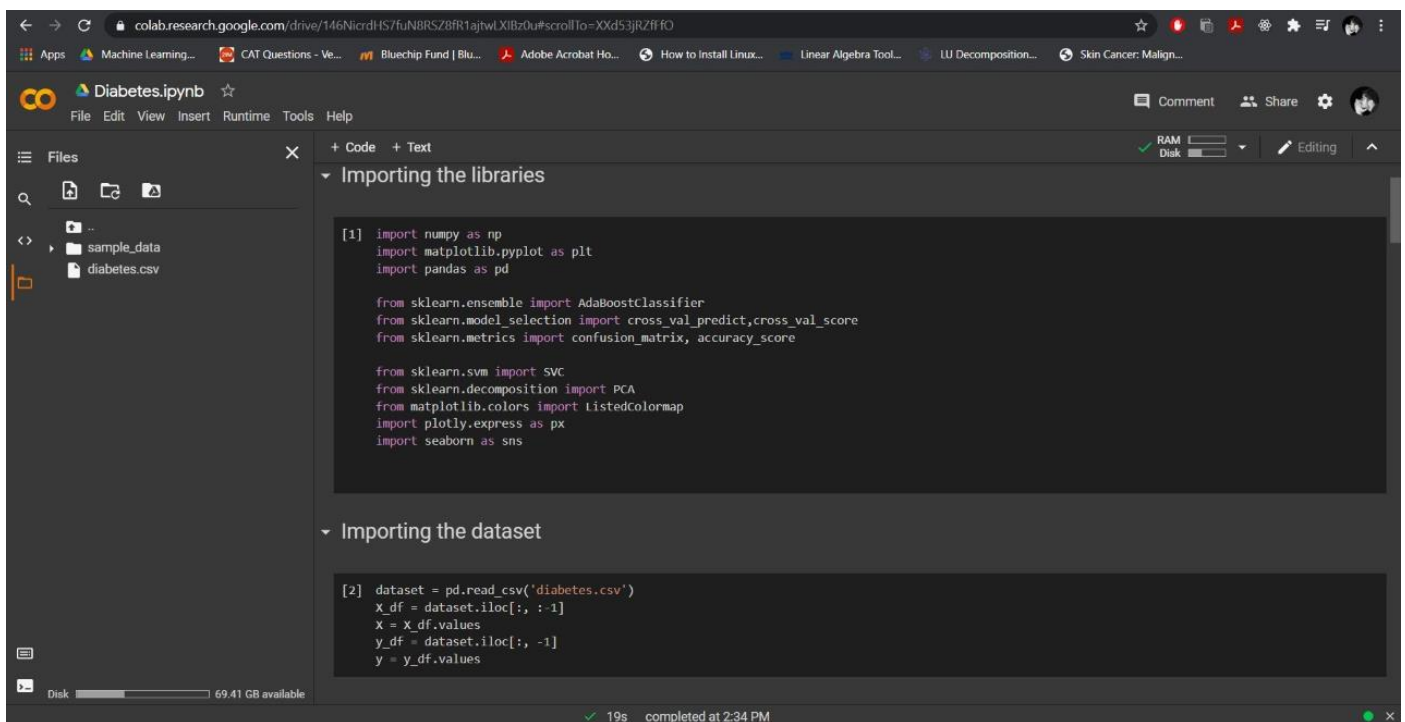
This piece of code compares the different independent variables and gives the importance of each feature by the means of an interactive bar graph. And as we can see through the result, some of the features have very less importance, so their absence won't alter the results much.



Diabetes Prediction –

Importing the Libraries –

In this screenshot, two cells of the code are shown, the first cell import all the libraries required for the project. The second cell however is used to read the test, and furthermore segregate the dataset in independent variables and dependent variable.



```
[1] import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_predict, cross_val_score
from sklearn.metrics import confusion_matrix, accuracy_score

from sklearn.svm import SVC
from sklearn.decomposition import PCA
from matplotlib.colors import ListedColormap
import plotly.express as px
import seaborn as sns
```

```
[2] dataset = pd.read_csv('diabetes.csv')
X_df = dataset.iloc[:, :-1]
X = X_df.values
y_df = dataset.iloc[:, -1]
y = y_df.values
```

Missing Data –

All the cells in this screenshot are parts of data pre-processing, as mentioned, the first cell specifically replaces the missing data of a particular column by the mean value of that particular column. The second cell splits the dataset into training test and testing set. The third cell comprises of two things, it is being used to reduce the dimensions of the given dataset to have a visual graphic second part performs standard scaling.

```
[3] from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])

[4] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.10, random_state = 0)

[5] pca = PCA(n_components=2)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_pca = sc.fit_transform(X_train_pca)
X_test_pca = sc.transform(X_test_pca)
```

Training kernel SVM on the Training Set -

The first cell in the code is the demonstration of the model being fitted on the dataset. This particular piece of code uses matplotlib library to plot the classification. The two axes represent the result obtained from dimensional reduction from pca code. The green area shows the positive result (1) and the red area shows the negative (0) result. This first plot is made off the training set and the second piece of code does the same for the test set and plot shows the result predicted by the machine.

colab.research.google.com/drive/146NcirdtIS7fuNBRSZBFR1ajtwXIBzOu#scrollTo=XXd53jRZffFO

Diabetes.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- diabetes.csv

Code Text

Training the Kernel SVM model on the Training set

```
[6] classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train_pca, y_train)

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
max_iter=1, probability=False, random_state=0, shrinking=True, tol=0.001,
verbose=False)
```

Visualising the fitting on training set

```
X_set, y_set = sc.inverse_transform(X_train_pca, y_train)
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 5, stop = X_set[:, 0].max() + 5, step = 0.25),
np.arange(start = X_set[:, 1].min() - 5, stop = X_set[:, 1].max() + 5, step = 0.25))
res = classifier.predict(sc.transform(np.array([X1.ravel(), X2.ravel()])).T).reshape(X1.shape)
plt.contourf(X1, X2, res,
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Kernel SVM (Training set)')
plt.xlabel('PCA Feature 1')
plt.ylabel('PCA Feature 2')
plt.legend()
plt.show()
```

19s completed at 2:34 PM

Visualisation on Test Set -

colab.research.google.com/drive/146NcirdtIS7fuNBRSZBFR1ajtwXIBzOu#scrollTo=XXd53jRZffFO

Diabetes.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- diabetes.csv

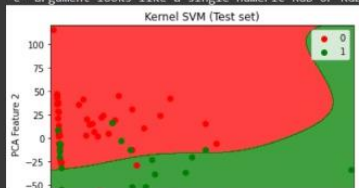
Code Text

Visualising the fitting on test set

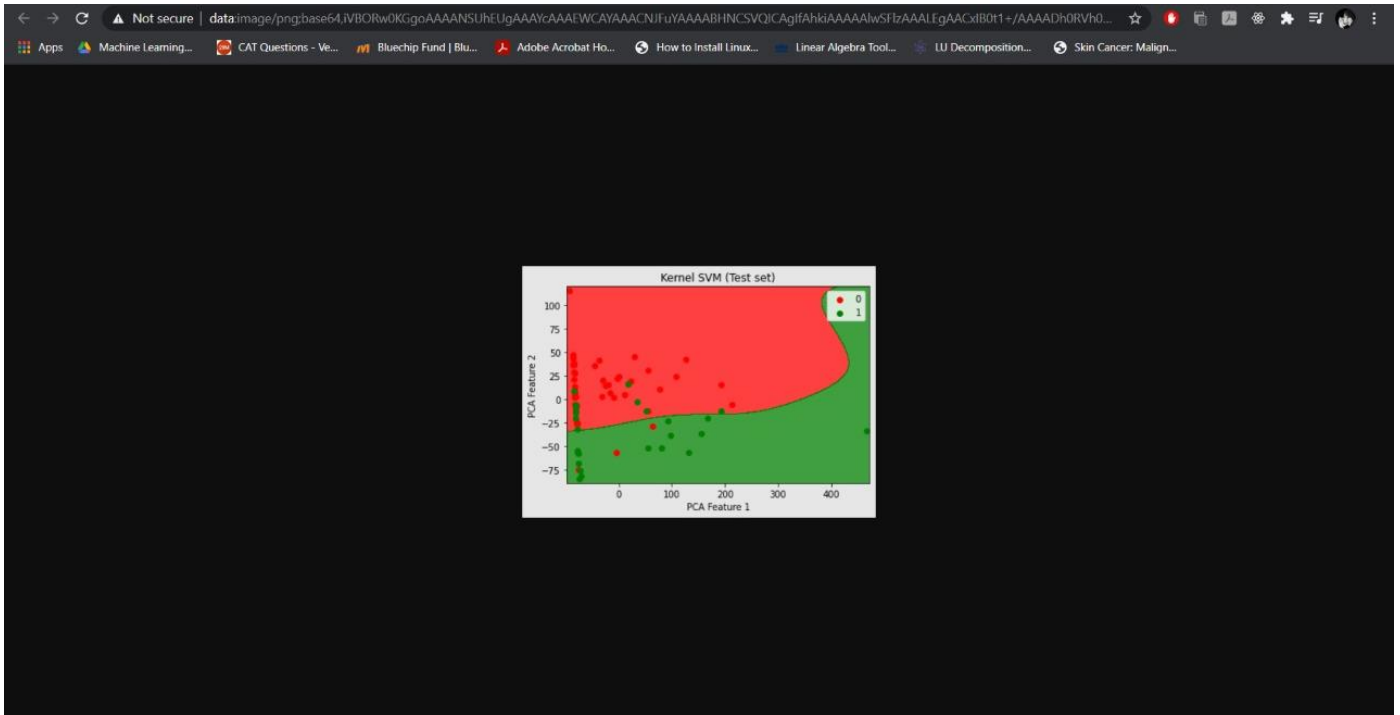
```
X_set, y_set = sc.inverse_transform(X_test_pca, y_test)
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 5, stop = X_set[:, 0].max() + 5, step = 0.25),
np.arange(start = X_set[:, 1].min() - 5, stop = X_set[:, 1].max() + 5, step = 0.25))
plt.contourf(X1, X2, classifier.predict(sc.transform(np.array([X1.ravel(), X2.ravel()])).T).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Kernel SVM (Test set)')
plt.xlabel('PCA Feature 1')
plt.ylabel('PCA Feature 2')
plt.legend()
plt.show()
```

c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length !=

c argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length !=

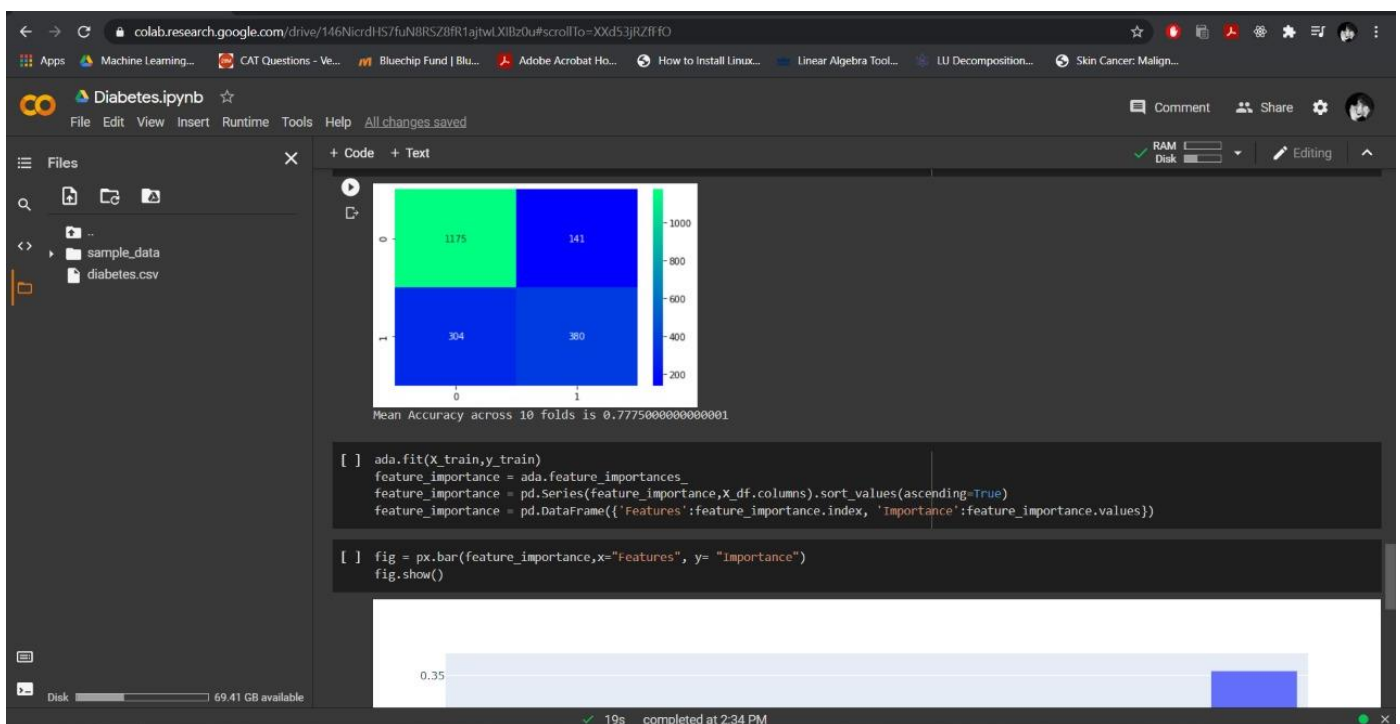
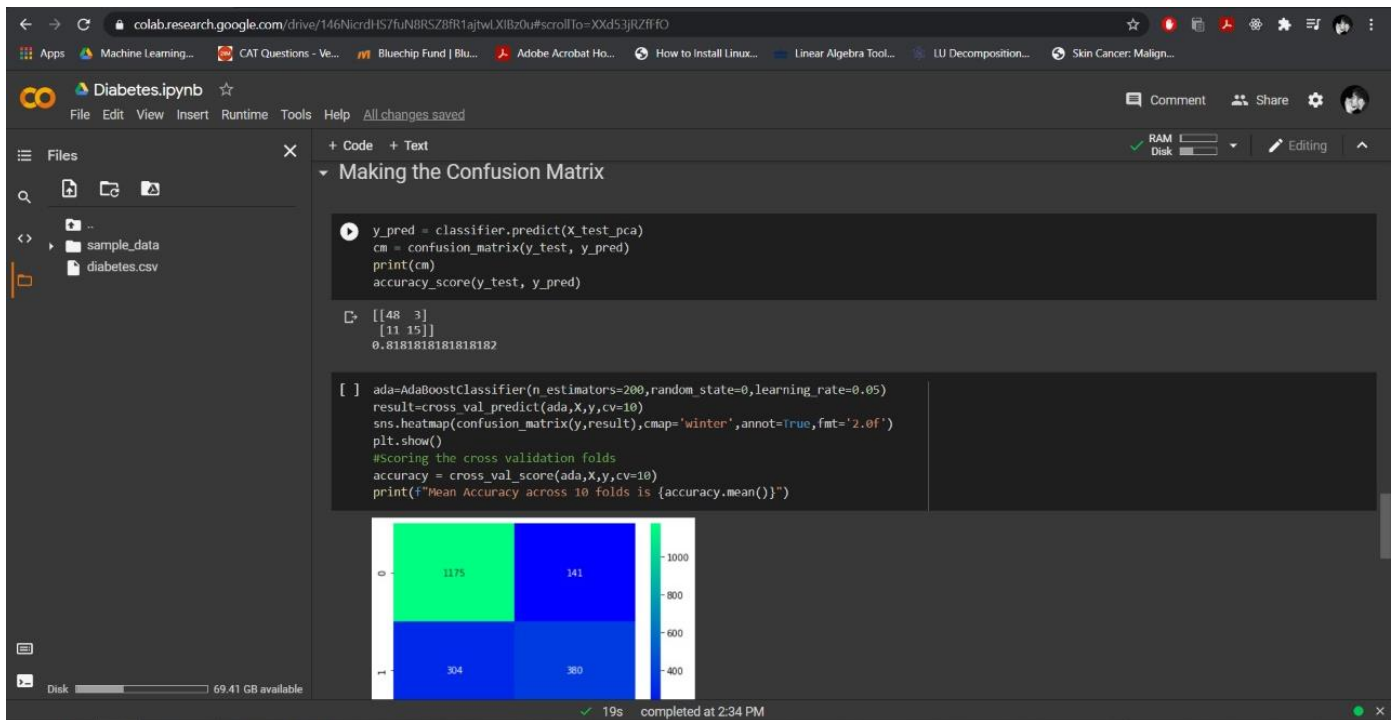


19s completed at 2:34 PM



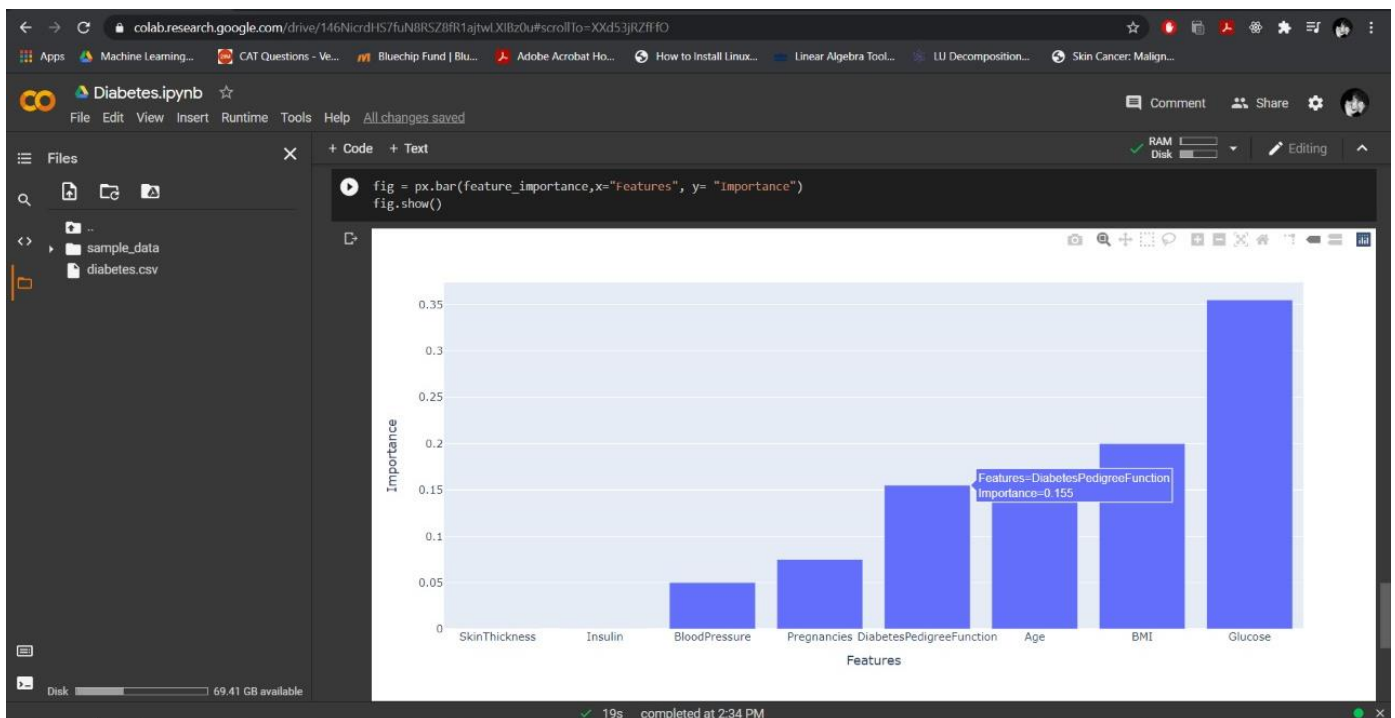
Making of Confusion Matrix –

This piece of code makes the confusion matrix and gives us the accuracy of the result which shows a promising 82 percent and there is interactive confusion matrix of all the learnings using adaboost.



Graphical Representation –

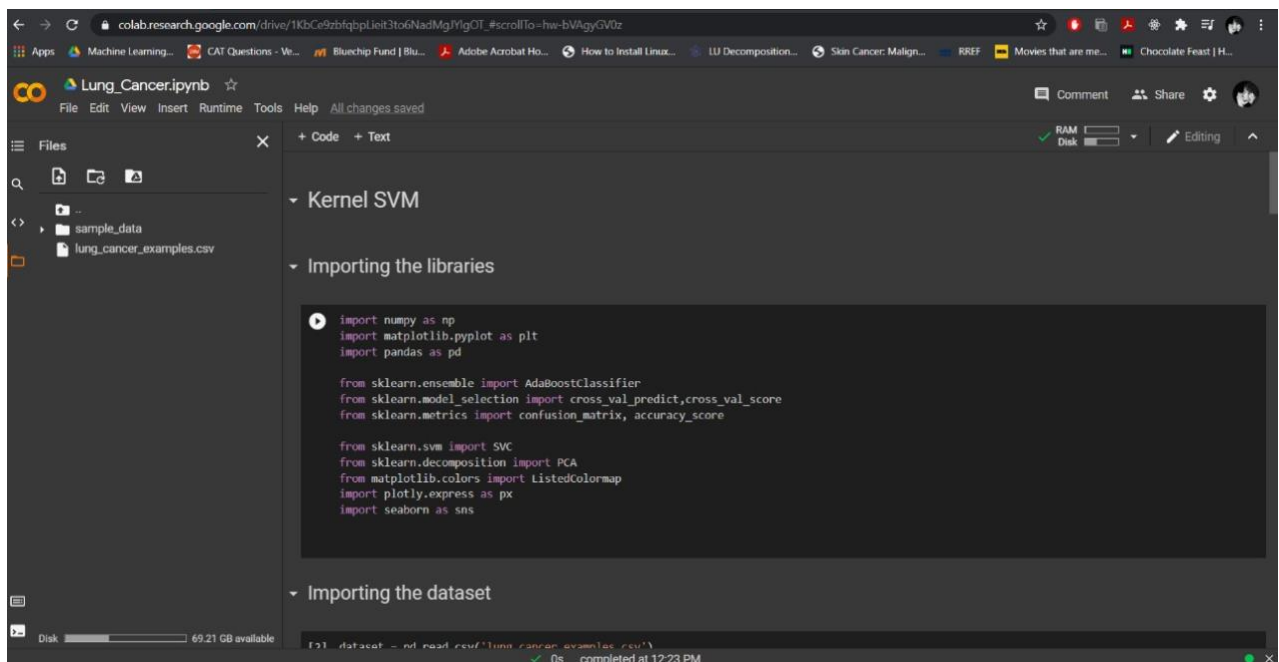
This piece of code compares the different independent variables and gives the importance of each feature by the means of an interactive bar graph. And as we can see through the result, some of the features have very less importance, so their absence won't alter the results much.



Lung Cancer

Importing the Libraries –

In this screenshot, two cells of the code are shown, the first cell import all the libraries required for the project. The second cell however is used to read the test, and furthermore segregate the dataset in independent variables and dependent variable.



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_predict, cross_val_score
from sklearn.metrics import confusion_matrix, accuracy_score

from sklearn.svm import SVC
from sklearn.decomposition import PCA
from matplotlib.colors import ListedColormap
import plotly.express as px
import seaborn as sns

dataset = pd.read_csv('lung_cancer_examples.csv')
```

Missing Data -

All the cells in this screenshot are parts of data pre-processing, as mentioned, the first cell specifically replaces the missing data of a particular column by the mean value of that particular column. The second cell splits the dataset into training test and testing set. The third cell comprises of two things, it is being used to reduce the dimensions of the given dataset to have a visual graphic second part performs standard scaling.

```
[2] dataset = pd.read_csv('lung_cancer_examples.csv')
X_df = dataset.iloc[:, 2:-1]
X = X_df.values
y_df = dataset.iloc[:, -1]
y = y_df.values

[3] from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(X)
X = imputer.transform(X)

[4] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Training kernel SVM on the Training Set -

The first cell in the code is the demonstration of the model being fitted on the dataset. This particular piece of code uses matplotlib lib library to plot the classification. The two axes represent the result obtained from dimensional reduction from pca code. The green area shows the positive result (1) and the red area shows the negative (0) result. This first plot is made off the training set and the second piece of code does the same for the test set and plot shows the result predicted by the machine.

```

colab.research.google.com/drive/1KbCe9zbfqplwE3to6NadMgMgOTI_#scrollto=hv-bVAgYGV0z
Machine Learning... CAT Questions - Ve... Bluechip Fund | Blu... Adobe Acrobat Ho... How to Install Linux... LU Decomposition... Skin Cancer: Malign... RREF Movies that are me... Chocolate Feast | H...
Lung_Cancer.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share Settings
Files
sample_data
lung_cancer_examples.csv
+ Code + Text
Feature Scaling
[5] pca = PCA(n_components=2)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_pca = sc.fit_transform(X_train_pca)
X_test_pca = sc.transform(X_test_pca)

Training the Kernel SVM model on the Training set
[6] classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train_pca, y_train)

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)

Visualising the result on training set
[7] X_set, y_set = sc.inverse_transform(X_train_pca), y_train

```

Visualisation on Test Set -

```

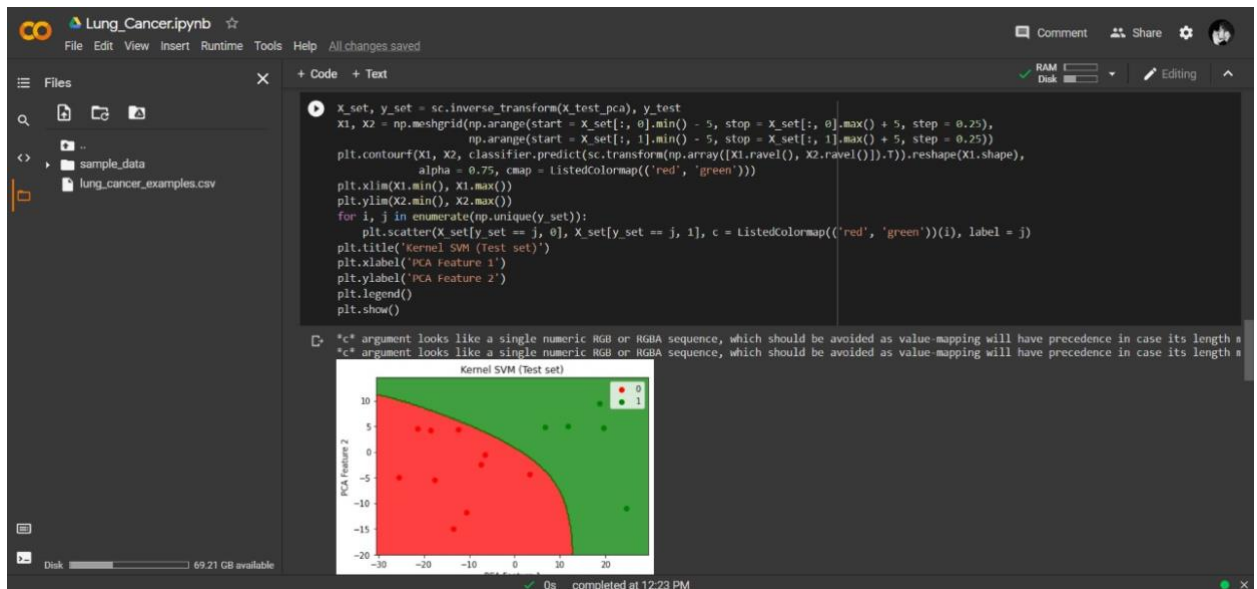
colab.research.google.com/drive/1KbCe9zbfqplwE3to6NadMgMgOTI_#scrollto=hv-bVAgYGV0z
Machine Learning... CAT Questions - Ve... Bluechip Fund | Blu... Adobe Acrobat Ho... How to Install Linux... LU Decomposition... Skin Cancer: Malign... RREF Movies that are me... Chocolate Feast | H...
Lung_Cancer.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share Settings
Files
sample_data
lung_cancer_examples.csv
+ Code + Text
Visualising the result on training set
[8] X_set, y_set = sc.inverse_transform(X_train_pca), y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 5, stop = X_set[:, 0].max() + 5, step = 0.25),
    np.arange(start = X_set[:, 1].min() - 5, stop = X_set[:, 1].max() + 5, step = 0.25))
res = classifier.predict(sc.transform(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape))
plt.contourf(X1, X2, res, alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Kernel SVM (Training set)')
plt.xlabel('PCA Feature 1')
plt.ylabel('PCA Feature 2')
plt.legend()
plt.show()

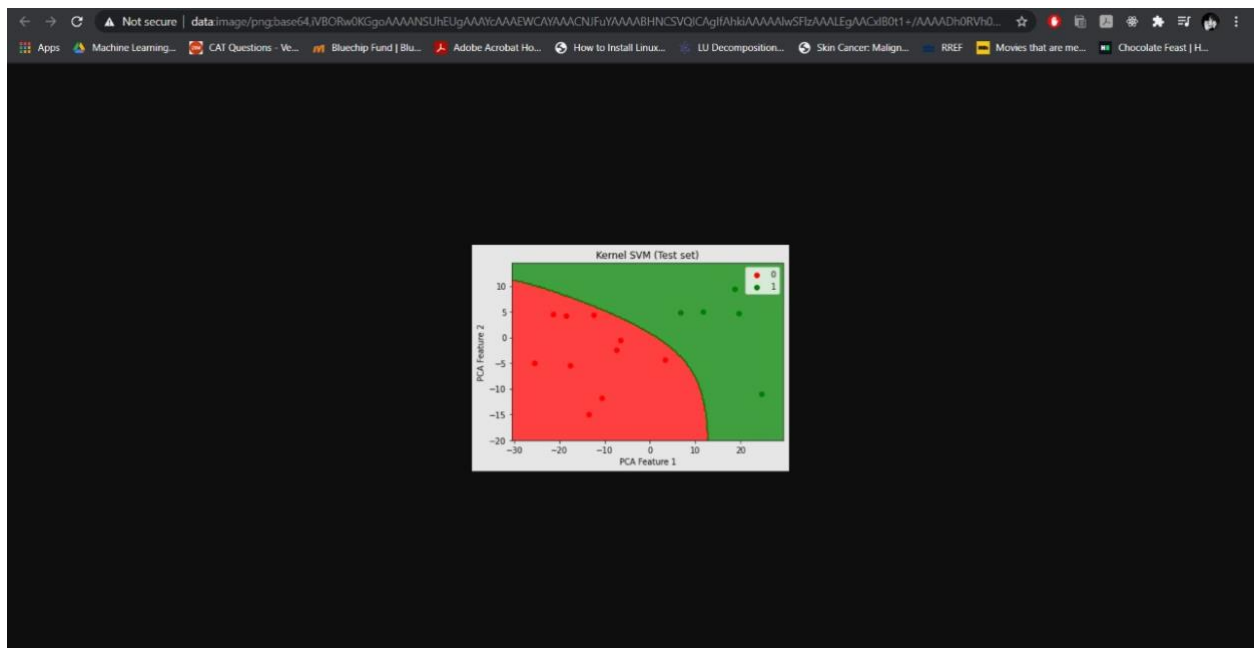
Kernel SVM (Training set)
PCA Feature 2
PCA Feature 1

```



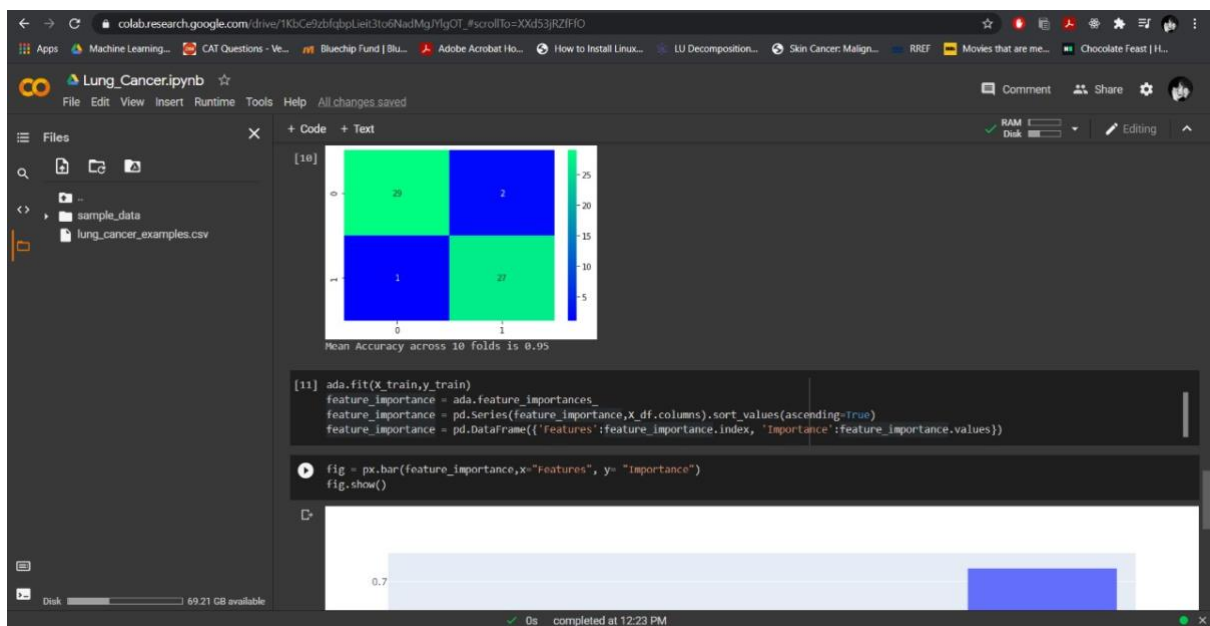
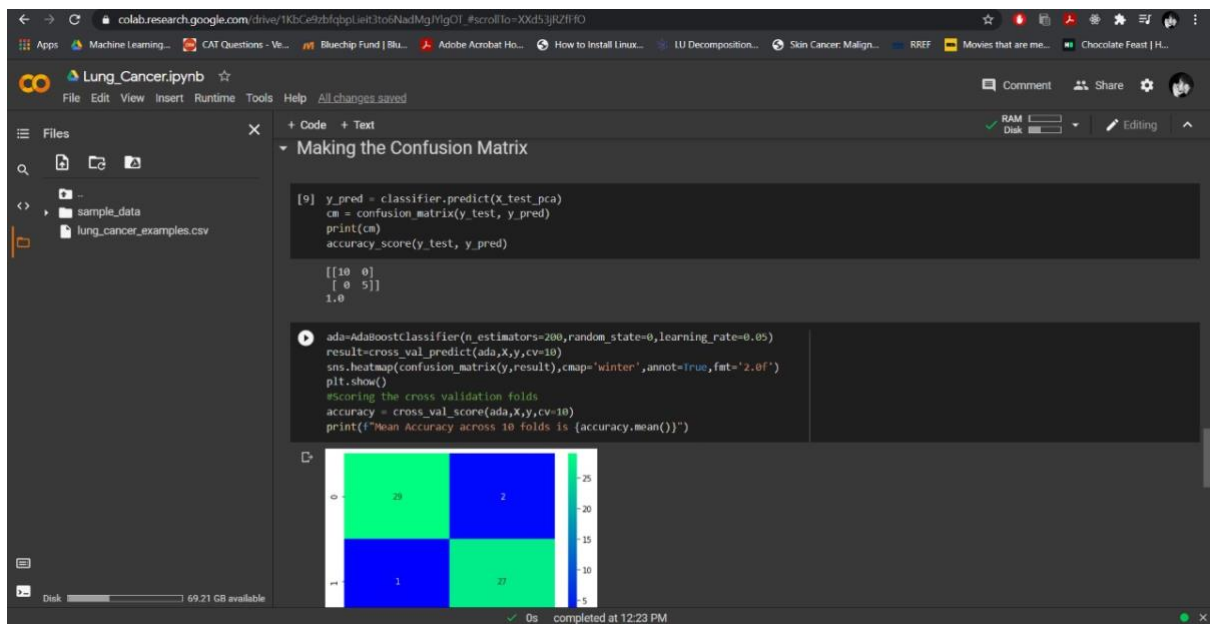
Test Set –





Making of Confusion Matrix –

This piece of code makes the confusion matrix and gives us the accuracy of the result which shows a promising 100 percent and there is interactive confusion matrix of all the learnings using adaboost.



Graphical Representation –

This piece of code compares the different independent variables and gives the importance of each feature by the means of an interactive bar graph. And as we can see through the result, some of the features have very less importance, so their absence won't alter the results much.

