CSCI 8456: Introduction to Artificial Intelligence

Final Project Report

Support Vector Machine and Multilayer Perceptron based Classifier for Cancer tissue classification

Girish Srinivas, Siddesh Southekal

Abstract:

Support Vector Machines (SVM) are one of the most popular machine learning algorithms. SVM is popularly used to model complex, real-world problems such as text and image classification, handwriting recognition, and in the areas of bioinformatics analysis. We apply Support vector machine based technique to classify tumor and normal tissue based on gene expression pattern obtained from RNAseq data.

We also implemented another machine learning algorithm, Neural Networks that attempts to mimic the learning pattern of natural biological neural networks using Artificial Neural Networks (ANN).

We tested these algorithms on Breast cancer dataset that were obtained from Cancer Genome Atlas (TCGA) [1] and tried to classify the data into two binary classes: Tumor and Normal samples based on attributes (expression values of tumor suppressor genes). The prediction accuracy of the algorithms were tested and compared with each other.

Introduction:

Data Set

Gene expression data of 1176 Breast cancer samples were downloaded from the NCI's Genomic Data Commons (GDC) [2] which is a unified data repository that enables data sharing across cancer genomic studies in support of precision medicine.

This data consisted of 99 Samples which were from Normal tissue and 1077 Samples from Tumor tissue. The Normal tissue samples are marked as negative (-ve) and tissue samples are marked as positive (+ve).

Test and Training Data:

The dataset for both positive and negative samples has been divided into Test and Training Data. The input data is represented as x*y matrix (where, x= number of tissue samples, y = number of features), where entry (x_i,y_j) represents the expression level of gene j in tissue i.

Each sample (i.e. matrix row) is associated with a +ve or -ve label indicating whether the sample tissue is cancerous or normal, respectively.

The data set has been divided in the ratio of 5 training: 1-test samples. The Training data consists of 898 +ve samples and 82 -ve samples. The test data consists of 179 +ve samples and 17 -ve samples.

Attributes:

The Breast cancer RNAseq data obtained from TCGA consists of 61,000 attributes. To eliminate non-informative genes and to further aid in computation, the number of genes/attributes have been reduced to 1217 genes to make it less computationally expensive. These 1217 genes (1018 protein-coding and 199 non-coding genes) belong to Tumor suppressor gene set. We decided to use the following 73 gene set with potential oncogenic role for our dataset (Table: 1). The data of only 70 attributes were available.

The list of genes were downloaded from Tumor Suppressor gene database [3]. The data was prepared using statistical computing tool R-3.4.3.

RHOA	BCR	FOXL2	RUNX1	CBL	CDKN1B	MAP3K8	DMBT1	DNMT1	DNMT3A
ECT2	ETS2	ETV6	EZH2	FOXO1	FOXO3	FLT3	FUS	GLI1	HDAC1
IDH1	FOXO4	MXI1	NOTCH1	NOTCH2	NOTCH3	NPM1	PAX5	PHB	PML
PTPN11	RARB	RB1	SKIL	SPI1	TCF3	WT1	ZBTB16	NR4A3	NCOA4
SPOP	KLF4	LITAF	YAP1	RASSF1	ARHGEF12	SIRT1	SUZ12	WHSC1L1	WDR11
SALL4	HOPX	LHX4	MIR106A	MIR107	MIR125B1	MIR146A	MIR150	MIR155	MIR17
MIR18A	MIR194-1	MIR194-2	MIR196A2	MIR20A	MIR203A	MIR210	MIR214	MIR222	MIR223
MIR24-1	MIR27A	MIR18B							

Table 1: List of 73 Tumor Suppressor genes with potential oncogenic role

Approach:

We used Python 3.6 for implementation of SVM classifier. Scientific computing software NumPy is also used. There are few SVM libraries available that we can use for implementation such as libSVM, SVM-Light, SVMTorch. Many general machine-learning libraries like scikit-learn also offer SVM modules, which are often wrappers around dedicated SVM libraries. The support vector machines in scikit-learn is used for training and testing the dataset. Neural Network implementation is also done using ScKit-Learn. The library matplotlib is used for generating the plots.

Support Vector Machines (SVM):

SVM is one of the most popular machine learning classification algorithm. SVM classifier is mostly used when there are more than two classification targets to predict. It is also used for regression and detection of outliers. SVM as a classification technique has many advantages, many of which are due to its computational efficiency on large datasets. The sparsity of the SVM solution leads to efficient data reduction for massive data at the testing stage, and the mathematical duality sanctions coherent handling of high dimensional data. This method uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. [4]

SVM Kernel

SVM is a kernel-based algorithm. A **kernel** is a function that transforms the input data to a high-dimensional space where the problem is solved. Kernel functions can be linear or nonlinear.

In scikit, different kernel functions can be specified for the decision function. We implemented different kernel types:

- Linear kernel
- Polynomial kernel
- Radial basis kernel (RBF)

The kernel values usually depend on the inner products of feature vectors for example inthe linear and the polynomial kernel, large attribute values might cause numerical problems and might need to be scaled. The kernel that we have implemented are briefly described below.

Linear Kernel:

$$K(xi, xj) = xT\ i\ xj$$

The dot product is used for linear SVM or a linear kernel because the distance is a linear combination of the inputs. The relation between the class labels and the attributes must be linear.

Polynomial Kernel

$$\boldsymbol{K(xi, xj) = (\gamma xiTxj + r)d, \gamma > 0}$$

Polynomial kernel can be used instead of Linear kernel. In Polynomial kernel represents the vector similarity (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

Radial Kernel

$$K(xi, xj) = exp(-\gamma kxi - xjk2), \gamma > 0$$

Radial Kernel nonlinearly maps samples into a higher dimensional space so it. radial kernel can create complex regions (polygons) within the feature space. RBF can handle the case when the relation between class labels and attributes is nonlinear.

*γ, r, and d are kernel parameters

Multilayer Perceptron:

We implemented supervised learning algorithm, Multilayer Perceptron (MLP) and tested it on our dataset. ANN are based on the idea of biological neural networks that have interconnected neurons with dendrites that receive inputs. Based on these inputs they produce an output signal through an axon to another neuron. Given the features, MLP therefore receives the

inputs and multiplies them by some weight. This is passed into an activation function (such as logistic function, a trigonometric function, a step function) to produce an output.

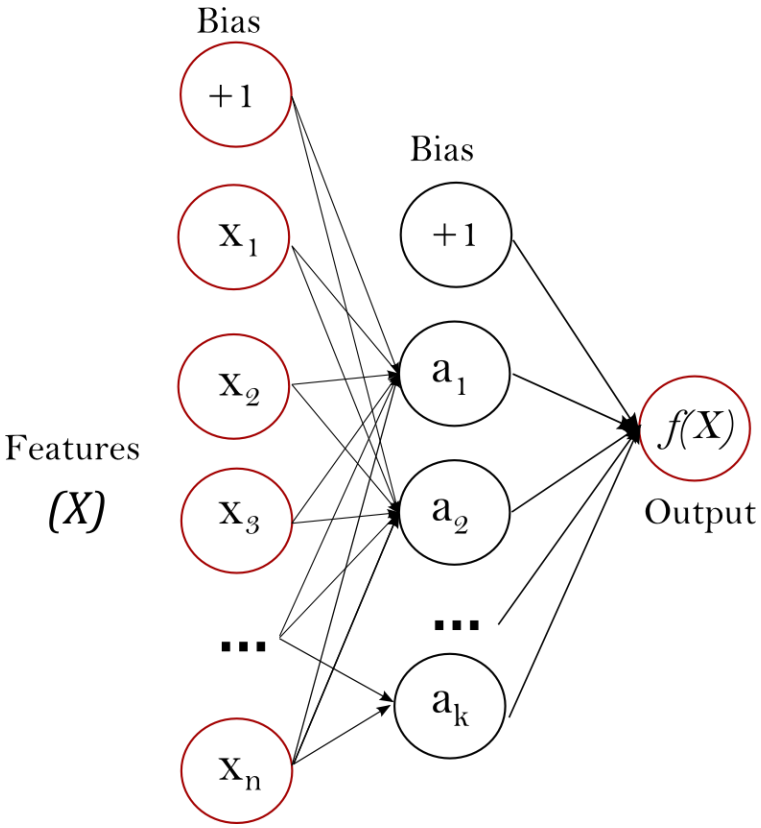


Fig1: Example of One hidden Layer MLP [5]

Confusion Matrix:

To Evaluate the performance of the classification model we calculated the Confusion matrix. The components of the confusion matrix are:

- True Positives (TP): Tumor Cases which were predicted correctly by the classification model
- True Negative (TN): Normal tissue samples which were predicted correctly by the model
- False Positive (FP): Normal tissue samples predicted as positive samples by the model

- False Negative (FN): Tumor samples predicted as normal samples by the model

- *Precision:*

Precision is the ratio of TP/ (TP+FP). Precision therefore indicates the ability of the classifier to classify the dataset correctly.

- *Recall:*

Recall/ Sensitivity is the ratio of TP/ (TP+FN) is the ability of the classifier to identify all tumor samples

Matthews Correlation Coefficient (MCC):

MCC is used to calculate the correlation coefficient between the observed and predicted values. The value of MCC is between -1 to +1. Where, a value of -1 indicates disagreement between observed and predicted values and +1 indicates perfect prediction. MCC is calculated using the equation below:

$$MCC = (TP * TN - FP * FN)/\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}$$

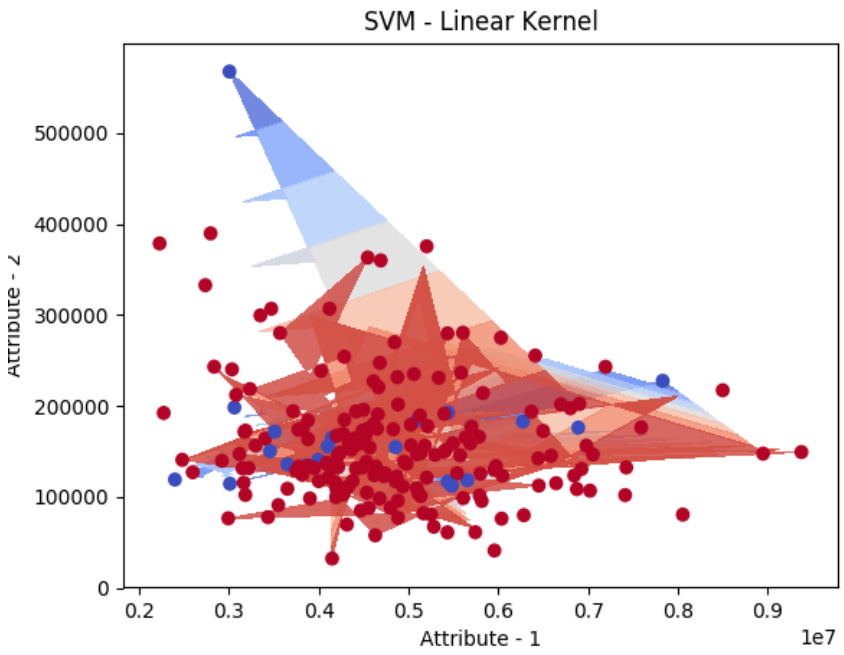
Results & Discussion

Algorithm Evaluation:

We trained the algorithm by considering different number of attributes and evaluated the prediction ability of the classifier. The optimum value of C and γ is not known initially. We tested the algorithm with different C and γ values to identify parameters that can accurately predict the test data. We tested our algorithm by altering these parameters to achieve better accuracy and there was not much difference in the prediction ability in the observed data set. It however, may not be useful to achieve high accuracy.

The user can choose the number of attributes to train the data set. For comparison, we trained the linear kernel for 20 attributes and for all the 70 attributes and compared the prediction accuracy. The confusion matrix is calculated for each and is displayed in the format Confusion matrix is calculated in the format [[TP FP ;FN TN]]. MCC is calculated for each run which gives a value between [-1 to +1].

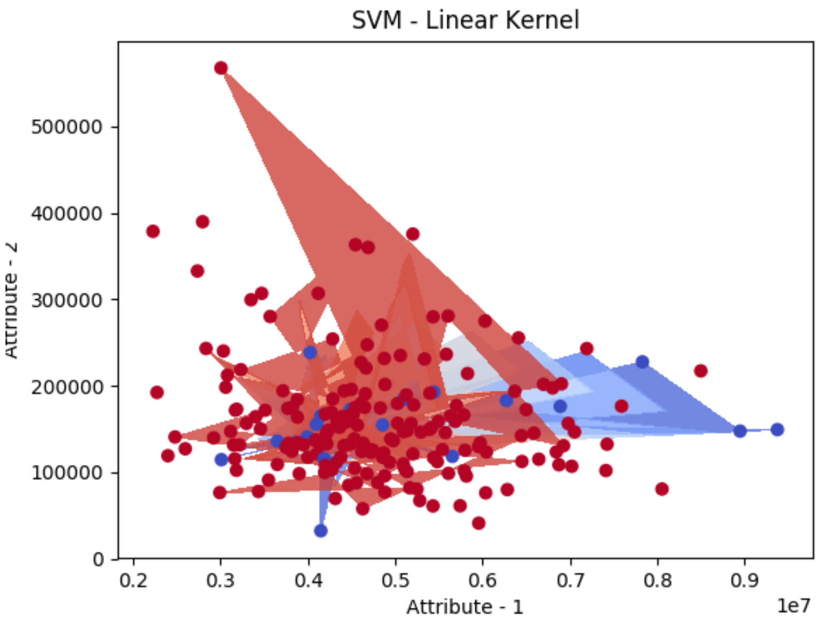
Linear Kernel with 20 attributes for training:



```
Using Linear Kernel
True Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
Predicted Classes:  [0 0 0 1 1 0 0 0 1 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1
 1 1 1 1 1 1 1 1 0 1 1]

Confusion matrix:
[[ 13   4]
 [  7 172]]

Classification Report:
             precision    recall  f1-score   support

          0       0.65      0.76      0.70        17
          1       0.98      0.96      0.97       179

avg / total       0.95      0.94      0.95       196


Correlation Coefficient:  0.674647260426
```

Fig 2 (a-b): SVM Linear Kernel using 20 attributes.

<u>Linear Kernel (all attributes):</u>

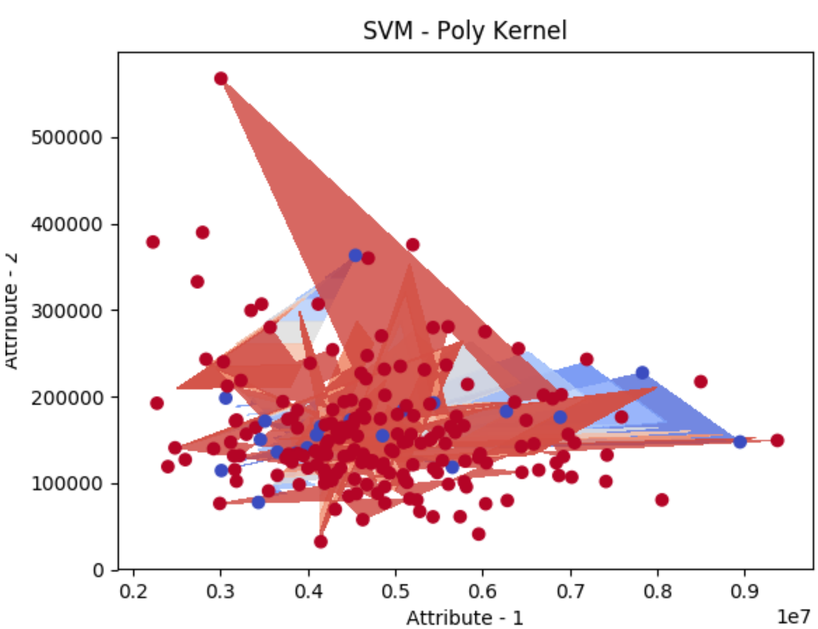


```
Using Linear Kernel
True Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
Predicted Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]

Confusion matrix:
[[ 17   0]
 [  2 177]]

Classification Report:
             precision    recall  f1-score   support

          0       0.89      1.00      0.94        17
          1       1.00      0.99      0.99       179

avg / total       0.99      0.99      0.99       196


Correlation Coefficient:  0.940606071858
```

Fig 3 (a-b): SVM Linear Kernel using all attributes

We observe that the algorithm accuracy significantly improved when we use more features for training the data.

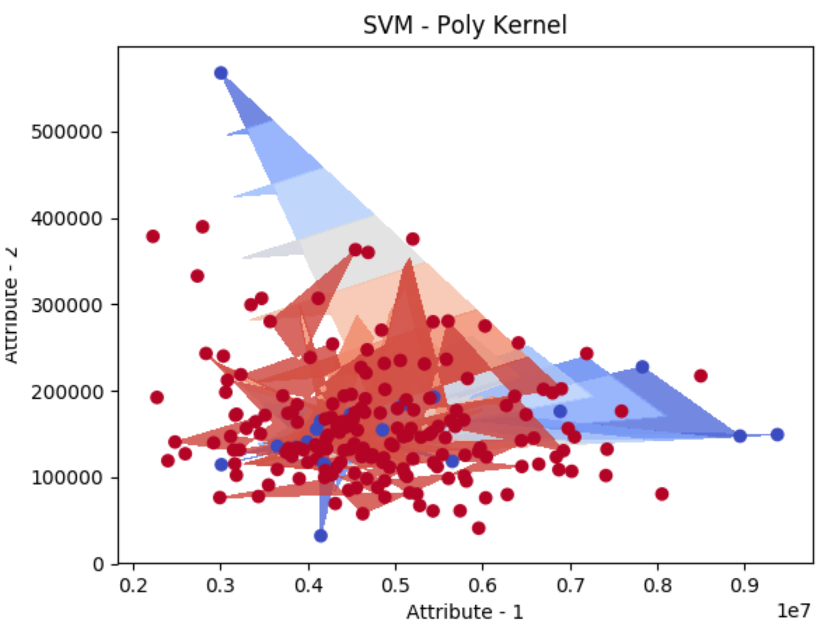
Polynomial Kernel (20 attributes):



```
Using Poly Kernel
True Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
Predicted Classes:  [0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 0 1 1]

Confusion matrix:
[[ 15   2]
 [  5 174]]

Classification Report:
             precision    recall  f1-score   support

          0       0.75      0.88      0.81        17
          1       0.99      0.97      0.98       179

avg / total       0.97      0.96      0.97       196


Correlation Coefficient:  0.794421592893
```

Fig 4(a-b): SVM Polynomial Kernel with 20 attributes

Polynomial Kernel (All attributes):

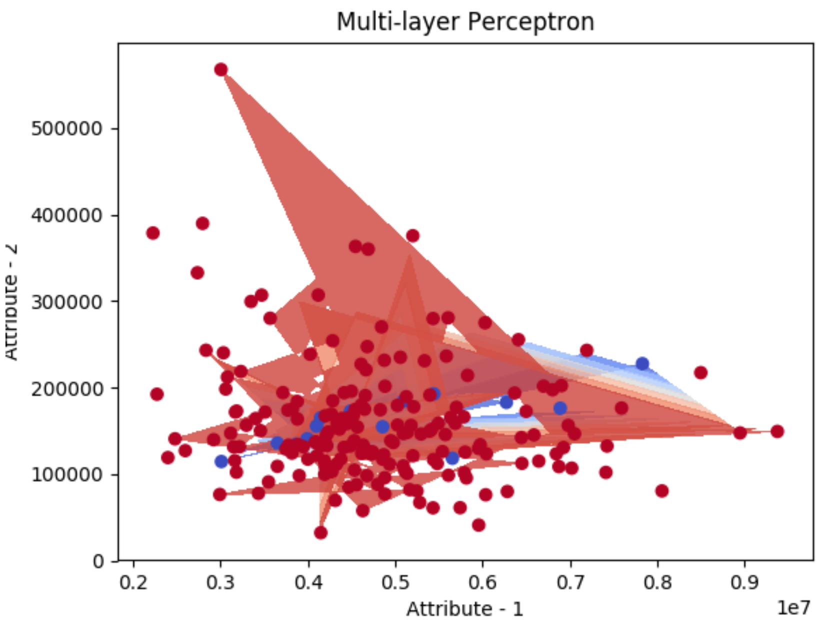


```
Using Poly Kernel
True Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
Predicted Classes:  [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]

Confusion matrix:
[[ 16   1]
 [  2 177]]

Classification Report:
             precision    recall  f1-score   support

          0       0.89      0.94      0.91        17
          1       0.99      0.99      0.99       179

avg / total       0.99      0.98      0.98       196


Correlation Coefficient:  0.906335947758
```

Fig 5 (a-b): SVM Polynomial kernel for all attributes

Multilayer Perceptron model was implemented using 3 layers and each layer has 10 neurons. (total 30 neurons) and tested it on our dataset. The prediction accuracy was much better than that of SVM. MCC value of 1 was achieved

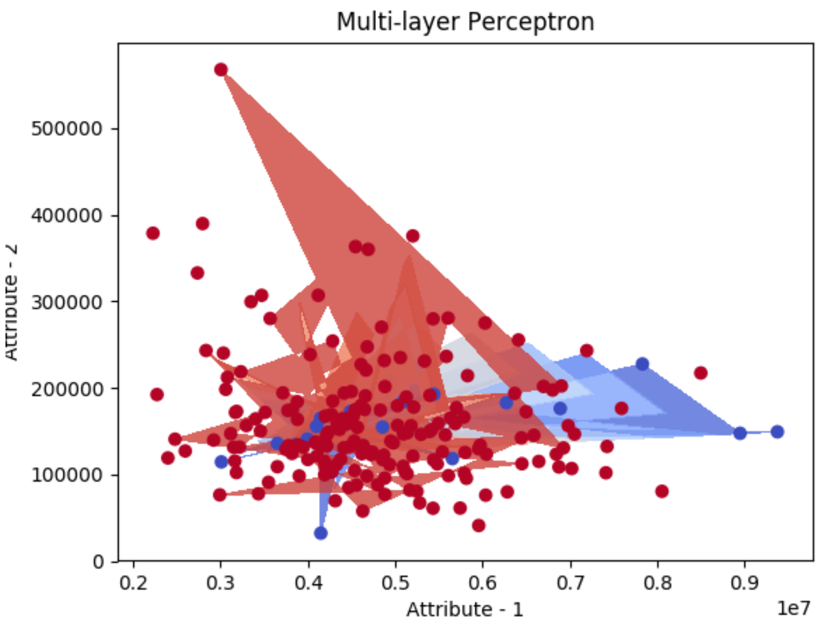
Multilayer Perceptron (20 attributes):




```
using Multi-layer Perceptron
True Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
Predicted Classes:  [0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]

Confusion matrix:
[[ 14   3]
 [  0 179]]

Classification Report:
             precision    recall  f1-score   support

          0       1.00      0.82      0.90        17
          1       0.98      1.00      0.99       179

avg / total       0.98      0.98      0.98       196


Correlation Coefficient:  0.899974861388
```

Fig: Multilayer Perceptron for 20 attributes

Multilayer Perceptron (All attributes):




```
using Multi-layer Perceptron
True Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]
Predicted Classes:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1]

Confusion matrix:
[[ 17   0]
 [  0 179]]

Classification Report:
             precision    recall  f1-score   support

          0       1.00      1.00      1.00        17
          1       1.00      1.00      1.00       179

avg / total       1.00      1.00      1.00       196


Correlation Coefficient:  1.0
```

Fig: Multilayer Perceptron for all attributes

Result Summary:

Matthews Correlation Coefficient (MCC):	SVM Linear Kernel	SVM Polynomial Kernel	Multilayer Perceptron
Attribute = 20	0.67	0.79	0.89
Attribute = 100	0.94	0.90	1.0

Table 2: This table summarizes the above result

Future Work:

We initially planned to run the above data on TensorFlow. Since the data set was < 5k we implemented multilayer perceptron to train the data set. Larger data set is required for running TensorFlow.

Program Instructions:

Please find the *Readme* file in the submission directory uploaded along with the files

References:

1. Home - The Cancer Genome Atlas - Cancer Genome – TCGA https://cancergenome.nih.gov
2. About the GDC https://gdc.cancer.gov/about-gdc
3. https://bioinfo.uth.edu/TSGene/
4. A Practical Guide to Support Vector Classication https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
5. 1.17. Neural network models (supervised) http://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron