International Journal of Engineering Applied Sciences and Technology, **2016**
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

Based On Protein Expression
Using Machine Learning Algorithms

Madhurya Manjunath Mamulpet
Faculty of Engineering, Environment and Computing, Coventry University
MSc Data Science and Computational Intelligence (ECT104) Stage 1
Coventry, United Kingdom
mamulpem@uni.coventry.ac.uk

Abstract- In this paper, the main objective is to perform classification on breast cancer proteins based on the family history, genes, cellular arrangements to identify the subtype of breast cancer. Breast cancer occurs when the cells are outgrown. This project uses Data from thousands of breast cancer protein and clinical patients that are diagnosed for the same and apply classification algorithm using Python programming language with the use of scientific libraries.

Keyword -Classification; machine learning; breast Cancer Tumor type; genes; clinical patient; python.

I. INTRODUCTION

The data set collected by a health organization which is approximately 80 patients record and 12.5k protein genes that are potential features that are diagnosed to proteins genes which causes breast cancer and few who do not have breast cancer. In this model, we find out the specific genes that are responsible for causing particular breast cancer type that can help medical practitioners to help prescribe medication more effectively and efficiently focus on treating for particular type rather in standardized medication.

The data set is very hard to be found as it is hard to identify the protein causing breast cancer and hence the dataset related to it, as there are more than thousands of proteins that need to be analyzed by the information retrieval methods, that presents the similarity between the protein genes which could belong to a particular subtype of breast cancer tumors.

This paper is organized in the following way:

- Chapter II literature review
- Chapter III Brief description of the dataset used in the classification analysis
- Chapter IV methodology applied used from machine learning
- Chapter V The results of experimenting on a dataset with the conclusion

II. LITERATURE REVIEW

The paper is published by Iman Rezaeian, Yifeng Li, Martin Crozier, Eran Andrechek, Aliounegom, Luis Rueda, and Lisa Porter this paper analysis the biological meaningful genes using support vector machine (SVM) classifier and feature selection for predicting the subtype of breast cancer tumor. The authors also propose various methodologies by the 5 types of subtypes Lumina A, Lumina B, HER2-enriched, Basal-Like, Normal-like. Each of the subtypes has its own characteristic features, the

objective is how to improvise the therapy approach by identifying the exact type and treating for that particular breast cancer type and not in general. The main emphasis on identifying the minimal number of genes that cause breast cancer. The use of 10-fold cross-validation is used to evaluate the accuracy of the model. Since few genes have are irrelevant uses the hierarchical decision-tree to produce the prediction for the type Lumina A and Lumina B. Now the new subtype contains 18 genes.

TABLE I. Results Top 20 genes ranked by the Chi-Squared attribute evaluation algorithm.

Rank	Gene Name	Rank	Gene Name	Rank	Gene Name	Rank	Gene Name
1	FOXA1	6	THSD4	11	DACH1	16	ACOT4
2	AGR3	7	NDC80	12	GATA3	17	B3GNT5
3	CENPF	8	TFF3	13	INPP4B	18	IL6ST
4	CIRBP	9	ASPM	14	TTLL4	19	FAM171A1
5	TBC1D9	10	FAM174A	15	VAV3	20	CYB5D2

The clinical patient is used to feed the tree for prediction, each leaf in the tree represents a subtype. It is very difficult to identify subtype Lumina A and Lumina B among other subtypes by the previous study and this is the primary reason that the Lumina A and Lumina B appear at the bottom of the tree. Later these Lum A and Lum B can be removed to prevent from misclassification on the other tumor types.

The research paper “Cancer Diagnosis” presented by Santa Cruz Country Fair 2012. They have a different approach to classify the breast cancer subtypes by four different machine learning algorithms Naïve Bayes algorithm they make an assumption that each feature is independent and unique among all others for labeling the features ,KNN cross-validation to find the accuracy of result of whether the gene is a subtype of breast cancer or not ,Decision tree comparing the performance and classifying by Breast cancer type basal, Lumina and a colorectal cancer dataset, predicting if breast cancer has a mutation in a particular gene.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

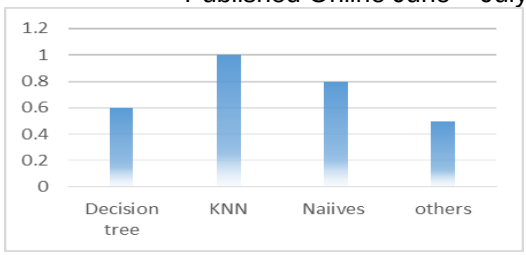


Fig1. **Sum of Performance correlation by methods used.**

It depicts the time taken vs the methodology applied out of which the KNN stands by far the best approach in analysis and performance.

Recently published book " Machine Learning Methods for Breast Cancer Diagnostic" By Shahnorbanun Sahran, Ashwaq Qasem, Khairuddin Omar, Dheeb Albashih, Afzan Adam, Siti Norul Huda Sheikh Abdullah, Azizi Abdullah, Rizuana Iqbal Hussain, Fuad Ismail, Norlia Abdullah, Suria Hayati Md Pauzi and Nurdashima Abd Shukor present a quantitative analysis about the clinical patient characteristic that can predict the accuracy of a person getting the breast cancer by image processing SVM test results that is much faster comparatively to other methods.

III. DATASET DESCRIPTION

The dataset used for this experiment is taken from an open source collection of metadata and the clinical data and protein data that are retained from a collection published in 2016.

This dataset contains information on breast cancer patients. It Includes 12553 unique genes from a total of 83 breast cancer patients. It also has clinical breast cancer dataset which includes clinical information from 105 breast cancer patients. The dataset also provides with a panel of genes, the PAM50 which is used to classify breast cancers into subtypes. Further examination has identified the variable 'Complete TCGA ID' to be in both the 77_cancer_proteomes_CPTAC_itraq dataset and the clinical_data_breast_cancer dataset. The Complete TCGA ID refers to a breast cancer patient, some patients can be found in both datasets. This is vital to our classification problem since there are clinical records and proteome data for each patient.

- The dataset contains
- 12,500 attributes that is the proteins
- 80 instances that are data about clinical patients
- In clinical data set it contains 30 features such as name age testing results and tested positive features from protein data
- The original clinical dataset contained 105 records due to misplaced values it was refined to 80 using a feature selection method

Research has identified 4 major molecular subtypes within breast cancer tumors, these subtypes are based on the genes that cancer cell express: Luminal A, Luminal B, HER2 Enriched, and Basal (triple negative). Currently, prediction and treatment decisions are based mainly on stage of tumor, grade of tumor, hormone receptor status and HER2 status. Molecular subtypes are mainly used in scientific research settings; they are, however, not part of a patients report and are not used to guide treatment. The use of molecular subtypes has greatly expanded, based on determining what genes are expressed in tumour samples, identifying subtypes of tumours can improve prognosis. Identifying and studying these subtypes has potential in planning more effective treatment and developing new therapies.

Gene characteristics include all information that makes a protein and by identifying which protein cell is making is abnormal it can tell about the size and growth. Though it is difficult to narrowly evaluate the protein level, Alternatively biologists calculate the level of RNA that acts as a connector between the genes and ribosomes that links to DNA

According to the research, the data were used to assess how the mutations in the DNA are affecting the protein expression landscape in breast cancer. Genes in our DNA are first transcribed into RNA molecules which then are translated into proteins. Changing the information content of DNA has an impact on the behaviour of the proteome, which is the main functional unit of cells, taking care of cell division, DNA repair, enzymatic reactions, and signalling etc. It performed K-means clustering on the protein data to divide the breast cancer patients into sub-types, each having unique protein expression signature. It found that the best clustering was achieved using 3 clusters (original PAM50 gene set yields four different subtypes using RNA data).

IV. EXPERIMENTAL SETUP

The dataset has been refined, yet it contains missing values of proteins in the patient record. As the dataset contained 300 approximately in missing proteins n patients record. The best ways to deal with the missing values are to:

Interpolation

We can assume data is missing in a specific way and use some form of interpolation. In this case each state/level of gene is recorded in terms of integer count of protein level is on time series then we can use linear interpolation between the level of one point of protein that we know and the level of other point protein and we assume the at centre is somewhere in the middle is a stable level.

We can assume there is a probability distribution over each of the unknown level of state. In this case, there are two levels values which we know and the in-between values that are not known and the values that fall in between are assumed to be sort of normal distribution. The variance increases quadrantally at each time point. So after the first

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

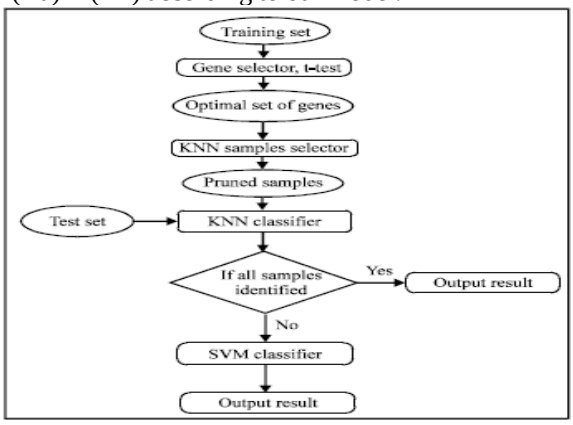
known value will have a Gaussian distribution and a next value will be even more bigger distribution and we take the sample of each backward that reaches a point. We define this level as a transition level

Here, feature selection is done by medical research out of which the proteins dataset is reduced to 50 features by research study the protein that are responsible for breast cancer have been identified.

The key usages of scikit libraries by window10, python 3.5 version in a Jupyter notebook, usage of programming and libraries

KNN

Its one of the simplest algorithm, when the model is built with knowing it is sorting the training dataset and when predicting for new samples the classifier looks at closest k nearest point. So, K actually defines the number of neighbours points used to make a decision by default it uses 5 neighbours. Assuming that instance locates to nearest protein cells and assign F(Xa)=F(Xn) according to our model.



If the level discrete then it returns the most common values of F among the K if it is consecutive then

$$F(Xa) = \sum_{i=1}^{k} \frac{F(xi)}{k}$$

To be precise the distance between each cell that triggers a particular tumor type can be evaluated by X attributes point Xi is the Euclidean distance

$$D(Xi, Xj) = \sqrt[2]{\sum_{F=1}^{Xi} [Xif - Xjf]^2}$$

- K point is assigned with a value to represent all the nearest neighbouring points. Where K point is the tumor type and nearest point are protein cells.
- New samples(X) will be given as input i.e in our
  case new samples of patients.
- Evaluating the distance between the X samples and identifying the K-nearest training data of (X).
- X is assigned to the same class as that of K nearest neighbours

Random Forest

It is a collection of Decision trees in a machine learning context. How Random forest is better than single decision tree has good prediction capacity but it is prone to overfeeding on part of data combining multiple trees retains the predictive power and it can reduce overfeeding by averaging the results afterwards to build the random forest classifier the major parameters is to specify the n estimator which refers to how many trees to implement the random forest classifier in sckitlearn for breast cancer data set when random state is set to 0 the accuracy of the train and testing data set is 87 which is a good result

It is an Ensemble machine learning algorithm that uses divide and conquer approach

The proposed algorithm

- Assuming the number the number of patient record set to N then sample of these N cases is taken at random.
- As there is more number of features in our case we set a constant variable m while the variables are selected such that m<M i.e. M is all the input features and m being random selected features.
- Each tree is grown to largest extent possible without pruning
- Predict new data by aggregating the prediction of the tree (i.e. average majority of classification)

Advantage

- The benefit of this method is that it can handle missing values and maintains accuracy for missing data.
- It won't over fit the model
- It handles dataset with higher dimensionality.

However, we could possibly obtain a better result by adjusting the parameter in random forest called max features which controls the randomness of each tree or we could apply pre-pruning which are similar to what is done in single decision tree set to parameter feature is important which helps to understand the way each features carries decision-making process since we are dealing with multiple trees randomness implies to each of them.

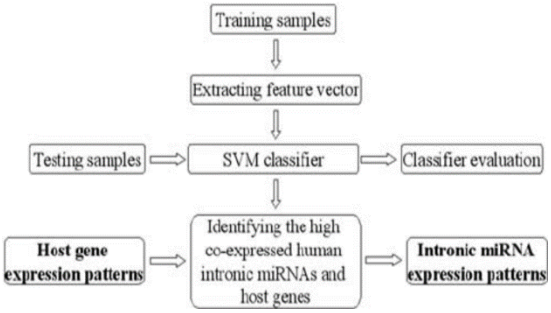
Support Vector Machine SVM

SVM is used for both classification and regression it is applied on linear and non-linear problems In this model we use svc support vector classifier to classify the type of breast cancer tumor. It looks for largest points n side of decision lines and these are support vectors the classification, in this case, maybe more accurate because you add a layer of complexity or requirement from the model for decision making. The idea is that the points need to be either very close to the decision line or very far to the division line. In our case, it is a linear representation of data.

International Journal of Engineering Applied Sciences and Technology, **2016**
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

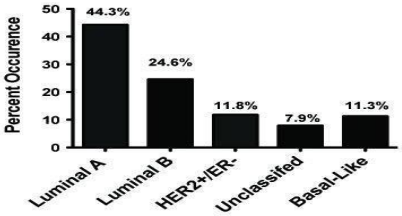
The goal is to design a hyperplane in other words classifying into groups. The best choice will be hyperplane that leaves the maximum margin from both classes

The kernel function is used where the complexity of the problem is high and data is linearly separable where it adds multiple polynomial features at a very high degree this way it prevents the computational complexity or burden that comes along adding multiple features to data. The dependencies of the variable are always taken into consideration.



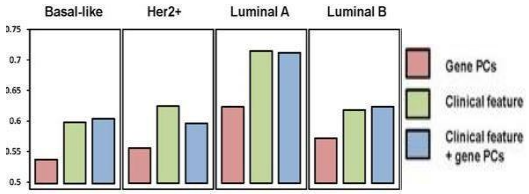
Advantages of support vector machine are

- High dimensional input space
- Sparse document vectors



- Regularization parameter
-

V. RESULT AND CONCLUSION



Method	The accuracy of the model	Remarks
KNN	75%	It was an efficient method but could not classify few proteins
Random Forest	92%	The best approach by far using the 10cross fold method
SVM	87%	it is a standardized method with a good accuracy prediction except for a small % of false positives

Table 2: Table of Results

The main goal was to obtain the proteins the belong to the four type of tumors those are the classes Lumina A Lumina B, HER-2 and Basil-like. The methods applied has given the efficient result. The best approach for this was SVM with an accuracy of 87% the only imitation/disadvantage was it false positives were high on the type Lumina A and Lumina B.

Figure 3: SVM Classification

A confusion matrix was generated between the type of tumor i.e. classes. To identify the false positives and found that the Lumina a was wrongly predicted with Lumina B.

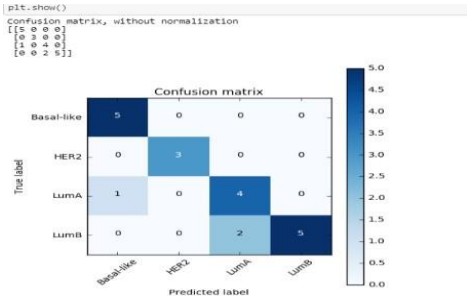


Figure 4: SVM Confusion Matrix

The KNN method approach gave one of the best results. With an accuracy of 75% classifying the protein into four types of the tumor but the disadvantages was due to missing values few of the proteins were not able to classify

The best fit for this approach was random forest where the missing values were dealt with the mutation method and use the 10 cross fold method. Where the decision tree was split to 10 decision trees with a higher accuracy of 92% on an average group. here where a certain protein that was not able to classify

CONCLUSION

The approach of Random Forest classifier is by far the best among all as the difficulty faced was on the missing value prediction of average using different approaches as a previously medical researcher has feature selected 50 among all it made it easy for applying the methodologies n to the existing data set.

The added advantage was the procedure had labelled classes which could help in choosing the most essential classes. Therefore, the methods approached were good enough and were obtained better results compared to the research. Further, the research can be carried out by classifying the dataset based on the patients age were the average of the clinical dataset can be taken to find the possible age type that could help people get diagnosed based on which they can prevent themselves before it gets worse.

REFERENCES

[1] S. A. Khan, D. He, and J. C. Valverde, "Pattern Recognition in Bioinformatics," *Biomed Res. Int.*, vol. 2016, no. June, pp. 1–2, 2016.

[2] F. M. Alakwaa, K. Chaudhary, L. X. Garmire, and B. G. Program, "Deep Learning Accurately Predicts Estrogen Receptor Status in Breast Cancer Metabolomics Data," 2017.

[3] A. Karplus, "Machine Learning Algorithms for Cancer Diagnosis,"

[4] *Compbio.Soe.Ucsc.Edu*, 2012.

[5] Y. Zhang, J. Xuan, R. Clarke, and H. W. Ressom, "Module-based breast cancer classification.," *Int. J. Data Min. Bioinform.*, vol. 7, no. 3, pp. 284– 302, 2013.

[6] S. Yepes and M. Mercedes Torres, "Mining Datasets for Molecular Subtyping in Cancer," *J. Data Mining Genomics Proteomics*, vol. 07, no. 01, pp. 1–7, Jan. 2016.

[7] E. Glaab, J. Bacardit, J. M. Garibaldi, and N. Krasnogor, "Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data," *PLoS One*, vol. 7, no. 7, 2012.

[8] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015.

[9] C. Nguyen, Y. Wang, and H. N. Nguyen, "Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic." 2013.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

APPENDIX

Import libraries

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

import itertools

from sklearn.grid_search import GridSearchCV
from sklearn import svm
from sklearn.svm import SVC
from sklearn.preprocessing import Imputer
from sklearn.cross_validation import train_test_split
from sklearn.cross_validation import KFold
from sklearn.pipeline import Pipeline
from sklearn.pipeline import make_pipeline
```

Loading data

In [ ]:
```
dataset_path = "77_cancer_proteomes_CPTAC_itraq.csv"
clinical_info = "clinical_data_breast_cancer.csv"
pam50_proteins = "PAM50_proteins.csv"

## Load data
data = pd.read_csv(dataset_path,header=0,index_col=0)
clinical = pd.read_csv(clinical_info,header=0,index_col=0)## holds clinical information about each patient/sample
pam50 = pd.read_csv(pam50_proteins,header=0)
```

In [2]:
```
merge = pd.read_csv('merge_proteome_clinic.csv', index_col=0)
```

In [3]:
```
proteome = merge.ix[:, :12553]
```

In [4]:
```
imputer = Imputer(missing_values='NaN', strategy='median', axis=1)
imputer = imputer.fit(proteome)
processed_proteome = imputer.transform(proteome)
```

In [5]:
```
X = processed_proteome
```

In [6]:
```
merge['PAM50 mRNA'] = merge['PAM50 mRNA'].astype('category')
merge['PAM50_mRNA_label'] = merge['PAM50 mRNA'].cat.codes
```

In [7]:
```
y = merge['PAM50_mRNA_label'].values
```

Splitting dataset

models and SVM works the best among traditional ML models)

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(features, y, random_state=42)

param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100], 'gamma': [0.001, 0.01, 0.1, 1, 10, 100]}
clf = GridSearchCV(svm.SVC(), param_grid=param_grid)
clf.fit(X_train, y_train)
```

Out[9]:
```
GridSearchCV(cv=None, error_score='raise',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
       fit_params={}, iid=True, n_jobs=1,
       param_grid={'gamma': [0.001, 0.01, 0.1, 1, 10, 100], 'C': [0.001, 0.01, 0.1, 1, 10, 100]},
       pre_dispatch='2*n_jobs', refit=True, scoring=None, verbose=0)
```

In [10]:
```
feature = SelectKBest(score_func=f_classif)
clf = svm.SVC()
pipeline = make_pipeline(
                         feature,
                         clf)

params_grid = {'selectkbest__k': [10, 20, 30, 40, 50, 100, 150],
               'svc__C': [0.001, 0.01, 0.1, 1, 10, 100],
               'svc__gamma': [0.001, 0.01, 0.1, 1, 10, 100]
              }

grid = GridSearchCV(pipeline,
                    param_grid = params_grid,
                    cv = 3)

grid.fit(X_train, y_train)
print('The final score of the grid object is', grid.score(X_test, y_test))
```

```
The final score of the grid object is 0.85
```

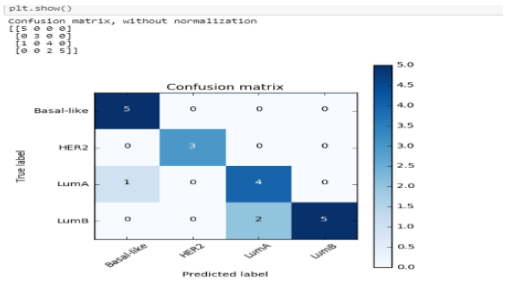
International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

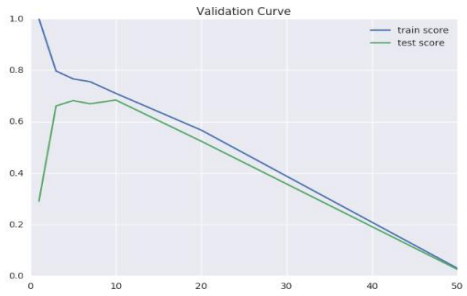
Creating classes i.e. tumor types

In [14]:
```
from sklearn.metrics import f1_score
f1 = f1_score(y_test, grid.predict(X_test), average=None)
classes=['Basal-like', 'HER2', 'LumA', 'LumB']
f1_result = dict(zip(classes, f1))
print('F1 score summary', f1_result)
```

```
F1 score summary {'Basal-like': 0.90909090909090906, 'HER2': 1.0, 'LumB': 0.83333333333333326, 'LumA': 0.7
2727272727272718}
```

In [12]:
```
# evaluate our model more thoroughly
import sklearn.metrics
from sklearn.metrics import confusion_matrix
cnf_matrix = confusion_matrix(grid.predict(X_test), y_test)
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

np.set_printoptions(precision=2)

fig = plt.figure(figsize=(5,5))
plot_confusion_matrix(cnf_matrix, classes=['Basal-like', 'HER2', 'LumA', 'LumB'])
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.show()
```





International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, No. 8, Pages 1-6
Published Online June – July 2016 in IJEAST (http://www.ijeast.com)

Knn classifier

In [53]:
```
# KN
# Use K-neighbor to get a general idea of how accurate the prediction could be.
from sklearn.cross_validation import cross_val_score, KFold
from sklearn.neighbors import KNeighborsRegressor
from sklearn.grid_search import GridSearchCV
from sklearn import svm

cv = KFold(n=len(X_p50), shuffle=True)

for n_neighbors in [1, 3, 5, 7, 10, 20]:
    scores = cross_val_score(KNeighborsRegressor(n_neighbors=n_neighbors), X_p50, y_p50, cv=cv)
    print("n_neighbors: %d, average score: %f" % (n_neighbors, np.mean(scores)))
```

```
n_neighbors: 1, average score: 0.290193
n_neighbors: 3, average score: 0.660622
n_neighbors: 5, average score: 0.681559
n_neighbors: 7, average score: 0.669095
n_neighbors: 10, average score: 0.683680
n_neighbors: 20, average score: 0.523664
```

In [54]:
```
# the difference between train score and test score is low, indicating low bias in the model
# However, the highest score is only around 70%, not high at all.
# I will talk about the reason at the end of this notebook.
from sklearn.learning_curve import validation_curve
n_neighbors = [1, 3, 5, 7, 10, 20, 50]
train_errors, test_errors = validation_curve(KNeighborsRegressor(), X_p50, y_p50, param_name="n_neighbors"
,
                                             param_range=n_neighbors, cv=cv)
plt.plot(n_neighbors, train_errors.mean(axis=1), label="train score")
plt.plot(n_neighbors, test_errors.mean(axis=1), label="test score")
plt.legend(loc="best")
plt.title('Validation Curve')
plt.show()
```