International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
SJIF (2024): 6.623

An Empirical Study on the Impact of Dataset Characteristics and Hyperparameters on Machine Learning Model Performance

Sony Annem

Email: *annemsony.137[at]gmail.com*

Abstract: ***In this experiment we evaluated the performance of Naive Bayes, Support Vector Machine (classification and regression models), we tried to plot the decision boundary for linear kernel and rbf kernel and used Ensemble Learning methods such as Bagging and AdaBoost, worked on three types of datasets (.mat file format's) such as Spam, Voting and Volcanoes.***

Keywords: Python, Jupyter Notebook, Classification algorithms (Decision tree, Naive Bayes, Support Vector Machine), Support Vector Regression, Ensemble methods (bagging, boosting)

1. Introduction

Big Data refers to the large, complex and diverse data sets generated by various sources such as social media, e-commerce, and IoT devices. Machine learning is a subfield of artificial intelligence that enables computer systems to improve their performance on a specific task by learning from experience, without being explicitly programmed.

In the context of Big Data, machine learning algorithms can be used to identify patterns, make predictions, and provide insights from vast amounts of data. There was Supervised and Unsupervised algorithm which plays a key role. Coming to Supervised algorithms which contains a label data, some common examples of supervised learning algorithms include linear regression, decision trees, k-nearest neighbors, and support vector machines. Unsupervised algorithms is to train model without any class labels. Classification model tries to predict the correct label/category/class of a given input data and Regression for investigating the relationship between independent variables or features and a dependent variable or outcome to predict continuous outcomes. The history of machine learning dates back to the mid-twentieth century, In the year 1940's the first computer-based machine learning models were developed during this period, focusing mainly on rule-based systems. Today, machine learning is used in a wide range of applications, including image and speech recognition, natural language processing, fraud detection, and recommendation systems. As data collection and computational capabilities continue to grow, it is expected that machine learning will continue to play a vital role in many aspects of modern life. As the part of assignment-2 we came across some machine learning models, along with metrics used to evaluate the performance of the model. The data-set can be of balanced and imbalanced, accuracy measure cannot be suits for the imbalanced data, in this assignment we implemented other metrics apart from accuracy.

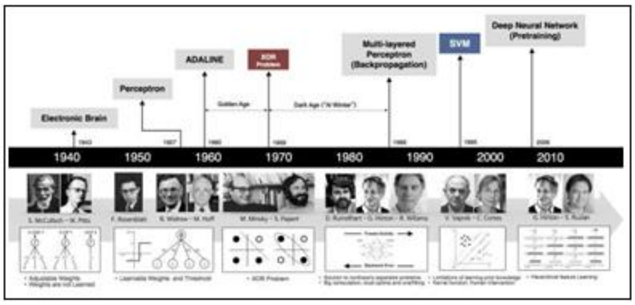


Figure 1: Concise History

2. Data-Set Details

We have received 3 types of dataset to work on it such as Spam, Volcanoes and voting. We divided the datasets into 70/30 ratio and my last four digits student ID (3093) as the random state seed for both data split and the method initialization.

a) About Spam Dataset: The dataset used to identify either sender spam or not spam. Below are the attributes:

1) Example index
2) geoDistance: Geographical distance between sender and ICSI, based on the MaxMind GeoIP database.
3) senderHour: The hour of packet arrival in sender's time zone.
4) Average IP Neighbor Distance: Avg. numerical dist. from sender's IP to the nearest 20 IPs of other senders of remote host as determined by p0f tool from SYN packet.
5) fngr_wss: Advertised window size from SYN received from remote host.
6) fngr_ttl: IP TTL field from SYN received from remote host.
7) OS: OS of remote host as determined by p0f tool from SYN packet. Windows, Solaris, Linux, UNKNOWN, FreeBSD, Others
8) pkts_sunk pkts sourced: Ratio of the number of packets sent by the local host to the number of packets received from the remote host.
9) rxmt_sourced: Approximate number of retransmissions sent by the remote host.
10) rxmt_sunk: Number of retransmissions sent by the local mail server.
11) rsts_sourced: Number of segments with "RST" bit set received from remote host.

12) rsts_sunk: Number of segments with “RST” bit set sent by the local mail server.
13) fins_sourced: Number of TCP segments with “FIN” bit set received from the remote host.
14) fins_sunk: Number of TCP segments with “FIN” bit set sent by the local mail server.
15) idle: Maximum time between two successive packet arrivals from remote host 3whs: Time between the arrival of the SYN from the remote host and arrival of ACK of the SYN/ACK sent by the local host.
16) jvar: The variance of the inter-packet arrival times from the remote host.
17) rttv: Variance of RTT from local mail server to remote host.
18) bytecount: Number of (non-retransmitted) bytes received from the remote host.
19) throughput: “bytecount” divided by the connection duration.
20) Class label (1 = spam, 0 = ham)

b) About Volcanoes Dataset: It contains images of venus planet in which 731 volcanoes and 1500 non-volcanoes. The data was collected by the Magellan spacecraft over an approximately four year period from 1990–1994, prediction occur based on the chip index. Ground Truth labels were obtained by experts, and contain labels indicating various degrees of certainty. To create a binary dataset, any instance with a certainty greater than or equal to 0.5 is positive; other are negative. The original labels indicate the location and radius of each candidate volcano in the radar image files (some included in ‘images’ folder). These volcanoes are extracted as 15-by-15 pixel “chips” around the center of each volcano. They were total of 15 chips for each image id, the chips are then flattened to form a 225-dimensional feature vector.

c) About Voting Dataset: The dataset consists of votes for a selection of popular bills taken from the GovTrack site. Voting records are shown for members of the U.S. House of Representatives who are either Democrat or Republican (to make the classification binary). Votes are given as either yea (’+’), nay (’-’), or ’0’ if the voter abstained.

3. Task 1: Na¨Ive Bayes Learner

As first part of task1, I used the volcanoes dataset to implement Multinomial Naive Bayes Method. First loaded the dataset into memory and splited it randomly into training and testing subsets in a 70/30 ratio. I used train test split function from the scikit-learn library to do this and used the precision score (), recall score (), and f1 score () functions to calculate the precision, recall, and F1 score, respectively. The algorithm works by calculating the conditional probability of each feature given each class label, and then using Bayes’ theorem to calculate the probability of each class given a new input. The class with the highest probability is then assigned to the input.

Normal each and every dataset can be of imbalanced and balanced, the metric measure were used to predict the model performance. Accuracy is a poor or misleading metric when dataset is imbalanced: positives or negatives are extremely rare. Accuracy measures the percentage of correctly classified in-stances over the total number of instances. It is a basic metric that gives an overall idea of how well the model is performing, but it can be misleading if the dataset is imbalanced. Precision measures the proportion of true positives (correctly classified positive instances) over the total number of positive predictions made by the model. It measures how often the model correctly identifies positive instances. High precision means that the model makes very few false positive errors. Recall measures the proportion of true positives over the total number of actual positive instances in the dataset. It measures how well the model can identify positive instances. High recall means that the model can correctly identify most of the positive instances.

Table I: Prediction Metrics Using Multinomial Naive Bayes Method

	Metric	Accuracy
1	Accuracy	82 %
2	Precision	73 %
3	Recall	67 %
4	F1-score	70 %

F1-score is the harmonic mean of precision and recall. It provides a balanced measure between precision and recall. A high F1-score indicates good balance between precision and recall.

From the above results we get to know that accuracy is not same with the precision values, accuracy is calculated by dividing the number of true positives(volcanoes) and true nega-tives(non volcanoes) by the total number of samples.Precision, on the other hand, measures the proportion of true positives out of all the samples that were predicted as positive by the classifier.

After sorting the precision, recall and f1score we get to know that f1 score lies in between the precision and recall, were the value of recall is 67% and precision is 73% then we got the f1score as 70% which is in between the recall and precision.

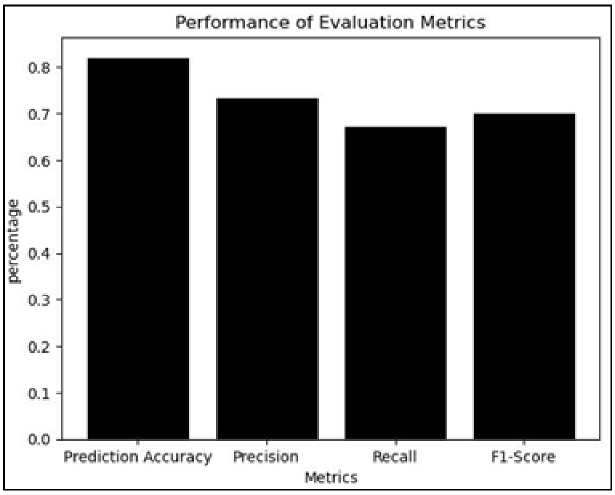


Figure 2: Plot Results

We observed from the result that f1 score lies in between recall and precision. For the given volcanoes dataset results the

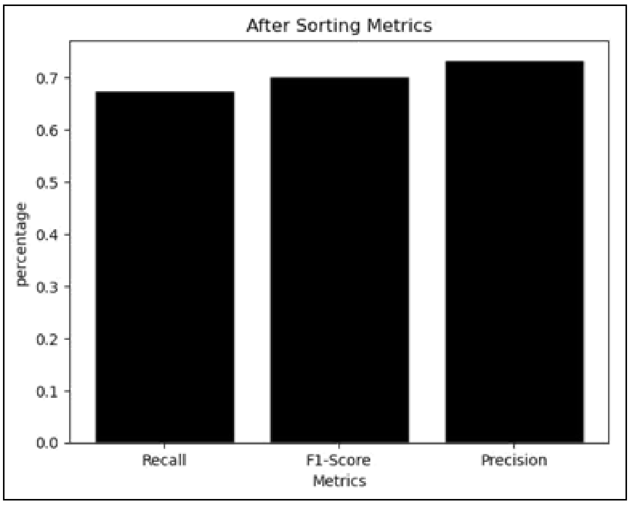


Figure 3: Metrics after Sorting the Scores

precision and recall values are not equal, then the F1 score will be a harmonic mean of the precision and recall values, and its position will depend on how the values compare to each other, here the precision is higher than the recall the f1-score will be closer to the recall and will be ranked lower than the precision value.

I used k fold (here k=5) cross validation technique to evaluate the performance of a model. The basic idea behind k-fold cross-validation is to split the dataset into k subsets or folds of roughly equal size. Then, the model is trained on k-1 folds and validated on the remaining fold. This process is repeated k times, with each fold being used as the validation set exactly once. But, unexpectedly the accuracy is same

Iris virginica and Iris versicolor). It consists of measurements of the sepal length, sepal width, petal length, and petal width for 150 iris flowers, each belonging to one of three species: Iris setosa, Iris versicolor, and Iris virginica. These measures were used to create a linear discriminant model to classify the species. The dataset is present in sklearn package, we need to use from sklearn.datasets import load iris. Hence, when –

Table IV: Using Gaussian Naive Bayes

	Data-set	Accuracy
1	Iris	95.56 %

comparing two models below it clearly shows that the accuracy of Iris dataset is higher than the Spam dataset. We know that Gaussian is applied to dataset with continues, meaning that the data follows a bell shaped curve. The features present in the spam dataset were not normally distributed, which can cause lower prediction. This is because the iris dataset is a well-

Table II: Overall Average Accuracy using 5-Fold

	Data-set	Accuracy after using 5-fold
1	volcanoes	82 %

as with and without k fold cross validation, it depends on factors such as size of volcanoes data set. As I build Gaussian Naive Bayes Classifier, were Gaussian Naive Bayes (GNB) is a probabilistic algorithm that uses Bayes' theorem with the "naive" assumption that all features are independent of each other given the class variable. GNB is particularly useful for high-dimensional datasets with continuous input variables. To build a Gaussian sklearn provides a beautiful package called Gaussian NB ().

And also we implemented on Iris data set were we know that Iris dataset contains four features (length and width of sepals and petals) of 50 samples of three species of Iris (Iris setosa,

Table III: Using Gaussian Naive Bayes

	Data-set	Accuracy
1	spam	37.41 %

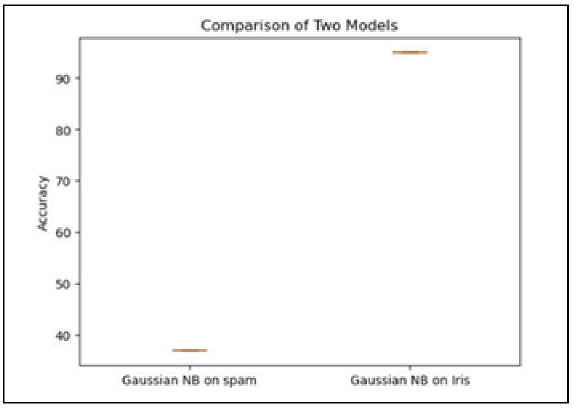


Figure 4: Comparison using Box plot

known benchmark dataset in machine learning with relatively clean and structured data, whereas spam datasets can be more noisy and complex. The main reason is the features in the Iris dataset are generally considered to be approximately normally distributed well suits for Gaussian.Given image shows the evidence of iris dataset which is in the shape of bell curve.

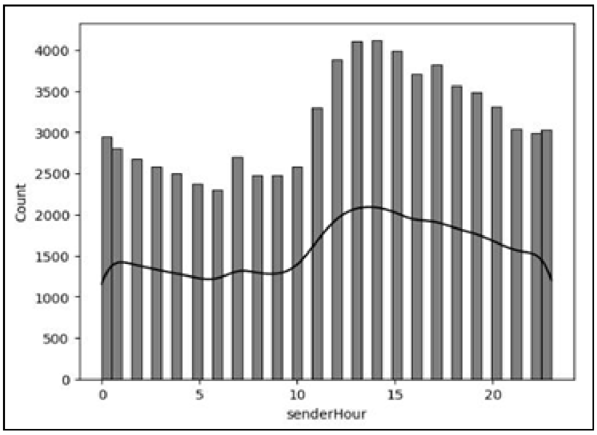


Figure 5: Spam Data Sample not in Normal Distribution (bell shape curve)

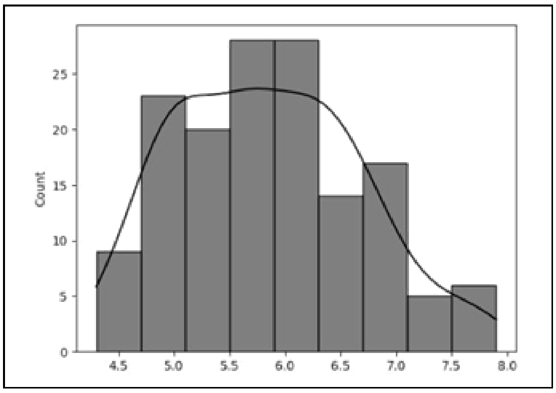


Figure 6: Iris Data Sample is in Normal Distribution (bell shape curve)

4. Task-2: Support Vector Machine

Support Vector Machine works on the hyperplane was proposed in 1960's by Dr. Vapnik and Dr. Chervonenkis. The algorithm tries to find the hyperplane that maximizes the margin between the two classes. The margin is the distance between the hyperplane and the closest data points from each class. The goal of SVM is to find the hyperplane that separates the two classes with the maximum margin. The data points closest to the hyperplane are called support vectors, using the SVC class from sklearn.svm. In SVM, the penalty parameter C is an important hyperparameter that controls the trade-off between maximizing the margin and minimizing the classification error on the training data.if C is too small, the model may underfit the data and have poor performance on the training and test data and also if C is too large, the model may overfit the voting data.

Table V: Prediction Accuracy on SVM

	Penalty Parameter	Accuracy
1	C=0.001	62.12 %
2	C=0.01	88.64 %
3	C=0.1	96.21 %
4	C=1	97.73 %

When C is small, the SVM model will choose a larger margin hyperplane even if some of the training examples are misclassified. This leads to a more generalized model with a wider margin but may cause some errors on the training set. When C is large, the SVM model will choose a hyperplane with a smaller margin, which will classify as many training examples as possible correctly. This leads to a more complex model that may overfit the training set Large Penalty Parameter leads to overfitting of data, it produce very complex or rigid classifier tries to fit all training datapoints, it does not correctly predict on new voting data.

In practice, an extremely large penalty parameter can result in a classifier that is overly complex and has poor generalization performance on new, unseen data. Then used SVM classifier and a Multinomial Naive Bayes classifier with default configurations using SVC and Multinomial NB classes from sklearn.svm and sklearn.naive bayes, respectively on voting dataset. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:1. True Positive Rate, 2. False Positive Rate. The AUC is a measure of how well a classifier can distinguish between positive and negative examples, which shows both accuracy scores are similar, with a value of 1 indicating perfect classification and a value of 0.5 indicating random guessing, in the example code, the cuve above shows that svm is ahead of binomial classifier with a higher AUC score indicating better ability to distinguish between positive and negative examples. From the above graph we get to know that SVM beats the Naive Bayes with little difference.

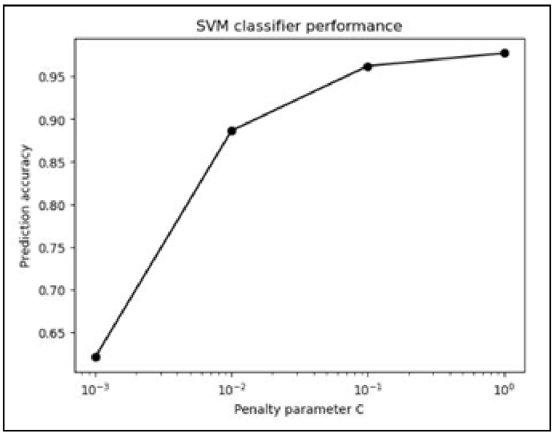


Figure 7: Effect Of Penalty Parameter on SVM

Table VI: Accuracy Using SVM and Naive Bayes

	Model	Accuracy
1	SVM	99.87 %
2	Naive Bayes	99.5 %

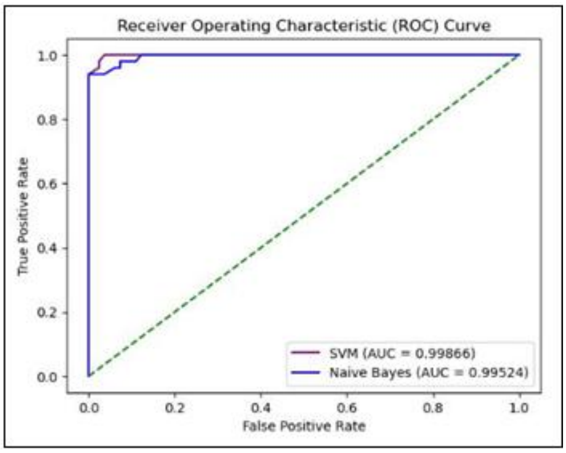


Figure 8: Comparison using ROC curve

The reasons were

1) Because SVM is more powerful model than Naive Bayes and can capture more complex decision boundaries between the classes.
2) Naive Bayes assumes that the features are conditionally independent, which may not hold true in practice. This can lead to a lower performance on voting data set where the features are correlated.

Continuation of the assignment we used only first two features of the dataset to train SVM classifiers with linear and rbfk-ernel. Meshgrids are used to create a dense grid of points in the feature space, which is then used to evaluate the classifier and visualize its decision boundary in a 2D space.

linear kernel is a simple kernel that performs a linear transformation of the input data into a higher-dimensional space. It works by computing the dot product between the input features and a fixed set of basis vectors in the feature space.

RBF kernel is a more complex kernel that performs a non-linear transformation of the input data into a higher-dimensional space. It works by computing the similarity between the input features and a set of basis functions, which are centered at each training sample.

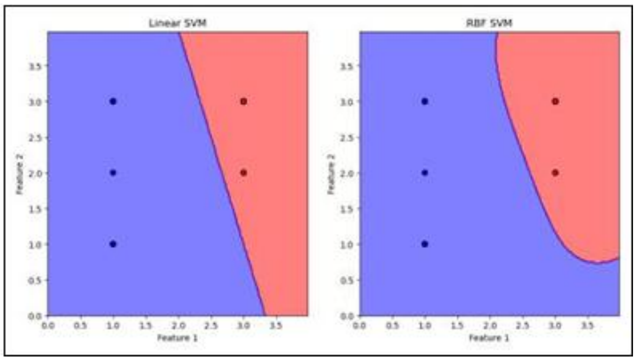


Figure 9: Decision Boundary using Linear and rbf kernel

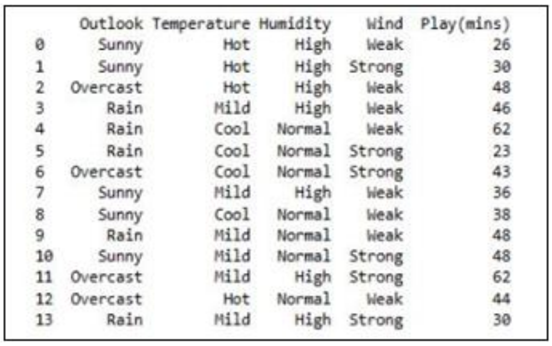
From the above two decision boundaries, we clearly observe that rbf SVM shows the good fit, The RBF kernel is more flexible than the linear kernel as it can map the input data into a higher-dimensional feature space, where it may be easier to find a linear decision boundary, the linear kernel is a good choice as it is computationally efficient and less prone to overfitting and RBF kernel may be a better choice as it provides more flexibility in finding a suitable decision boundary.

5. Task 3: Regression Analysis

Regression analysis is a statistical method that is used to model the relationship between a dependent variable and one or more independent variables

A. Table lists the 14 days of Jason playing tennis records. Use NumPy to represent the data as array

1) Dataset: It contains details about the player plaing the game in minutes. The attributes such as outlook, temoerature, humidity, wind used to measure the number of minutes the players were playing the game. We trained the above dataset using SVR, SVR is used for predicting continuous numerical values, and it is especially useful when dealing with non-linear relationships between the input variables and the output variable. I used SVR () to build the Regressor mode, the epsilon parameter, also known as the error tolerance parameter, is a key parameter that controls the width of the margin of tolerance for errors in the regression model.



```
     Outlook Temperature Humidity    Wind  Play(mins)
0      Sunny         Hot     High    Weak          26
1      Sunny         Hot     High  Strong          30
2   Overcast         Hot     High    Weak          48
3       Rain        Mild     High    Weak          46
4       Rain        Cool   Normal    Weak          62
5       Rain        Cool   Normal  Strong          23
6   Overcast        Cool   Normal  Strong          43
7      Sunny        Mild     High    Weak          36
8      Sunny        Cool   Normal    Weak          38
9       Rain        Mild   Normal    Weak          48
10     Sunny        Mild   Normal  Strong          48
11  Overcast        Mild     High  Strong          62
12  Overcast         Hot   Normal    Weak          44
13      Rain        Mild     High  Strong          30
```

Figure 10: Dataset

Table VII: Support Vector Regression Model

	Epsilon Value	R-squared score
1	ϵ=0.1	-0.85
2	ϵ=1	-1.02
3	ϵ=10	-1.06

The highest R-squared score is -0.8577296316038296, which is obtained with epsilon=0.1 In this case, the R-squared score for epsilon=0.1 is the highest, followed by epsilon=1, and epsilon=10. This is because a smaller epsilon value leads to a tighter margin in the SVR model and better fit to the data When epsilon is set to a smaller value, the margin of error is reduced, and the model tries to fit the data more closely, resulting in a higher R-squared score. On the other hand, when epsilon is set to a larger value, the margin of error is increased, and the model allows for more deviation between the predicted values and actual values, resulting in a lower R-squared score. A smaller epsilon value means that a smaller margin of tolerance is allowed, resulting in a tighter fit between the predicted values and the actual values. A larger epsilon value, on the other hand, means that a larger margin of tolerance is allowed, resulting in a looser fit between the predicted values and the actual values. The R-squared score, on the other hand, is a measure of how well the regression model fits the data. Small ϵ leads to larger R-square value in the same way larger ϵ leads to smaller R-square value. A larger epsilon value can help prevent overfitting and result in a model that is more robust and generalizes better to new data.

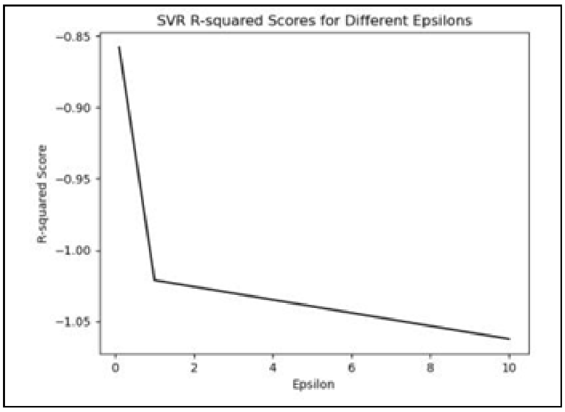


Figure 11: R-squared score based on ϵ value

Continuation of the assignment to build a K-Nearest Neighbors (KNN) Regression on the given dataset, were K-Nearest Neighbors (KNN) Regression is a non-parametric regression technique that uses the k closest data points in the training set to predict the value of a new data point. In KNN regression, the predicted value for a new data point is the average of the k nearest neighbors' values. To train a KNN Regression model using scikit-learn, I used the K Neighbors Regressor class from the sk learn. neighbors module. Selecting the value of k that generates the highest R-squared score is not always the best approach in K-Nearest Neighbors (KNN) Regression. For K=1 the R-square is -0.35 which is highest and for K=10 the R-squared is low -1.25 by observing we will know that smaller K value have best R-Squared but it leads to overfitting problem. while the R-squared score can be used to evaluate the performance of a KNN Regression model, it should not be the sole criterion for selecting the value of k.

Table VIII: K-Nearest Neighbors Regression Model

	K value	R-squared score
1	K=1	-0.35
2	K =5	-1.52
3	K =10	-1.25

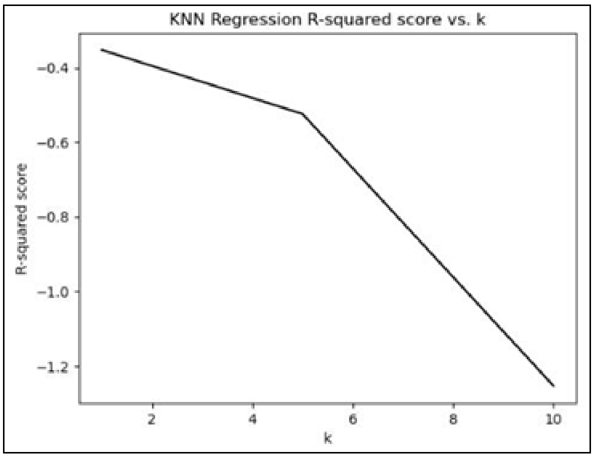


Figure 12: R-squared Score and K value

6. TASK 4: Ensemble Learning

For the given Volcanoes data set we compared linear model with two other ensemble models such as Bagging ensemble, AdaBoost ensemble using 5-fold cross validation. decision trees are a simple and interpretable method that can work well with small to medium-sized datasets with discrete or continuous features. Bagging generates multiple decision trees by sampling the training data with replacement, and the final prediction is made by taking the average of the predictions of all trees. AdaBoost iteratively trains decision trees on the training data, with each new tree focusing on the samples that were misclassified by the previous trees. The accuracy of Bagging model is high with 86% while Adaboost and linear classifier got the same accuracy. The reason might be that the AdaBoost can be prone to overfitting if the number of iterations is too high, or if the weak learners are too complex. Bagging, on the other hand, can handle overfitting by reducing the variance of the underlying model.

Table IX: Linear Vs Ensemble Accuracy

	Model	Accuracy
1	Decision tree	0.75
2	Bagging model	0.866
3	AdaBoost Model	0.75

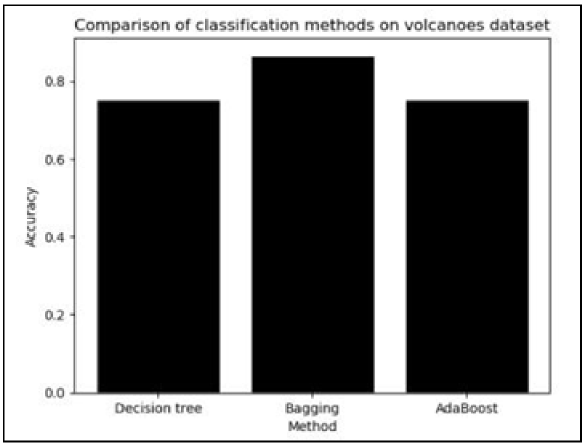


Figure 13: Linear Decision tree Vs Ensemble models on volcanoes dataset

Bagging method is well suits for this dataset because it predicts the value based on multiple tree, bagging method some times called random tree and by using ensemble method instead of a single learner. Their are some reasons to use ensemble instead of simple model:1. Improve the accuracy of the predictions by combining the outputs of multiple learners, each with different strengths and weaknesses. In this dataset ensemble model gives the best accuracy compared to simple, in some cases, a single learner may be sufficient or even preferable, especially if the dataset is small or the underlying model is simple.

7. Conclusion

Model performance can be effected by the size of dataset, as from task-1 I observed that F1-score lies in between precision and recall and after using 5-fold cross validation also their is no change in the accuracy. Gaussian NB got have accuracy for IRIS dataset than Spam dataset due to the data type, Iris data set is well formatted in Normal distribution. From task-2 I observed that the penalty parameter impose accuracy, the smaller C leads to under-fit and larger C leads to over-fit result in more accuracy. And also I observed that SVM beats Naivebayes with little difference on voting dataset, rbf kernel given the best decision boundary compared to linear kernel. From task-3 I observed that ϵ also impose the R-squared score, the smaller ϵ good R-squared score compared to larger ϵ and also smaller K good R-squared error but we should not selected k value based on R-score and finally in task-4 bagging model beats linear decision model and AdaBoost model with the accuracy of 86% and concluded that ensemble models are not always required for smaller dataset.

References

[1] https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce

[2] https://towardsdatascience.com/ensemble-models-5a62d4f4cb0c
[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear
[4] https://scikitlearn.org/stable/modules/linear model.html.
[5] https://www.geeksforgeeks.org/ensemble-methods-in-python.