AI4I 2020 Predictive Maintenance Dataset

1. Introduction

Artificial intelligence has become the most interesting technology and has huge potential in helping in any sector of human life whether it is for educational purposes or entertainment purposes or industrial purposes since nowadays data is available in abundance in any field where a person can run a simple model of machine learning and can get good predictions results in one of the applications where Artificial intelligence is in industries where in industries have expensive pieces of machinery on which they spent allot to buy them besides buying them the main concern of the managers is to maintain them because machine failure can cause a pause on whole production process since all the machines in industries are interlinked as we know every machine has downtime and can get fail at one particular moment

Artificial intelligence can help in predicting machine failure and this project attempts to predict machine failure which in turn helps the industry to reduce downtime and product quality. The data set that we are working on is a synthetic dataset that, to the best of our knowledge, accurately represents genuine predictive maintenance faced in the industry. Real predictive maintenance datasets are typically difficult to get and, in particular, difficult to publish. This aids in the real-time prediction of machine failure since all the examples are drawn from actual measurements such as air temperature, torque, and process temperature. This data is stored with actual values of sensors which are numeric in nature and it has no images or any other modes of data that mean data is not multimodal which is easy for us to work with one more thing which helps us and make it easy for us to work on this data is whether there the data is structured or unstructured. as structured data is made up of rows and columns of numbers and values while unstructured data is made up of sensors, text files, audio and video files, etc. This data is structured. In this preliminary phase, we were concerned with the quality of the data. By identifying duplicate data, any missing or undefined values, and any potential outliers An initial choice of the pertinent characteristics for the function of the work has to be accomplished

Dataset

The dataset consists of 10000 data points stored as rows with 14 features in columns UID: unique identifier ranging from 1 to 10000 product ID: consisting of a letter L, M, or H for low (50% of all products), medium (30%) and high (20%) as product quality variants and a variant-specific serial number air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K. rotational speed [rpm]: calculated from power of 2860 W, overlaid with a normally distributed noise torque [Nm]: torque values are normally distributed around 40 Nm with an if = 10 Nm and no negative values. tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process. and a 'machine failure' label that indicates, whether the machine has failed in this particular data point for any of the following failure modes are true. In below table 1.1, the description of attributes is given

Attribute name	**Description**
Product ID	Identification of a product of the process. It is characterized by a letter between L, M, and H which indicates the quality (low medium, or high) followed by a serial number.
Type	The type of product made. Coincides with the letter in the Product ID
Air Temperature [K]	Air temperature, in Kelvin, is achieved by generating random numbers in later processed to have one Gaussian distribution of mean 300 K e standard deviation of 2 K
Process Temperature [K]	Process temperature was generated randomly, normalized with a standard deviation of 1 K, and added to the air temperature plus 10
Rotational Speed [rpm]	Rotation speed calculated from a power of 2860 W and superimposed on a Gaussian noise
Torque [Nm]	Torque whose values are distributed normally around an average of 40 Nm and with a standard deviation of 10 Nm. All values are non-negative
Tool wear [min]	Wear of the machinery expressed in minutes
Machine Failure [binary]	Label showing if it a broken or not. '1' if it occurred, ‘0’ if not
tool wear failure (TWF)	The tool will be replaced or fail at a randomly selected tool wear time between 200 â€“ 240 mins
heat dissipation failure (HDF)	heat dissipation causes a process failure if the difference between air- and process temperature is below 8.6 K and the total rotational speed is below 1380 rpm.

Table 1.1 Description of variables

power failure (PWF)	If the product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails
overstrain failure (OSF)	If the product of tool wear and torque exceeds 11,000 minimum for the L product variant (12,000 M, 13,000 H), the process fails due to overstrain.
random failures (RNF)	Each process has a chance of 0,1 % to fail regardless of its process parameters.

2. Preprocessing

Every real data comes with some unknown random values or some missing values or irrelevant values because of human error in calculation or some sensors error in recording the value it is very important to identify those values in order to rectify it and make our model more reliable if those values are not been taken care of it can result in wrong predictions the whole process of transforming raw data into useful and efficient format is called preprocessing

Data Processing is done in three stages data cleaning, data transformation, and data reduction

I. Data cleaning:

In data cleaning, we have to deal with missing data and noisy data where the missing values are replaced by the mean of the whole data or the most probable value

In this data, the missing values were either empty values or represented by NA so we had to find both and replace them with mean the code which we used to find it is given below

```
ai4i.isnull().sum().any()    this finds Null
ai4i.isna().sum().sum()      this finds NA
```

Noisy Data is data that does not make any sense it can be either an extremely large value or an extremely small value which will result in our model getting trained wrongly or it can be unnecessary data we don’t need

In this data set, the duplicate columns were being removed since it has no use the code is given below

```
print ("Duplicate Rows:")
duplicate = ai4i[ai4i.duplicated
(subset=ai4i.columns.difference(['Machine
failure']),keep='last')]
```

II. Data Transformation

This involves the Normalisation of data, Attribute selection, Data Discretisation and many more. Tranformation of data is cruicial since many times data can be in different forms like in case of dealing with pictures data has to be transformed in meaningful way to use it for our intrest in our data set we performed data discretisation by doing one hot encoding on type as the values of type were high[H], low[L], medium[M] This results in encoding type in zeroes and ones which will help our model The code and example is given below

```
ai4i = pd.get_dummies(ai4i, columns = ['Type'])
```

Type H	Type L	Type M
0	0	1
1	0	0

III. Data Reduction

Having huge amount of data is good but having huge amount of data which we dont use is bad. Data reduction is a process to reduce and remove unnecessary data. Data reduction is done based on the requirements which involves attributes selection Dimensionality reduction Dimensionality reduction in this data set was done by Dropping columns Productid, TWF, PWF, OSF, RNF from the dataset the code is given below

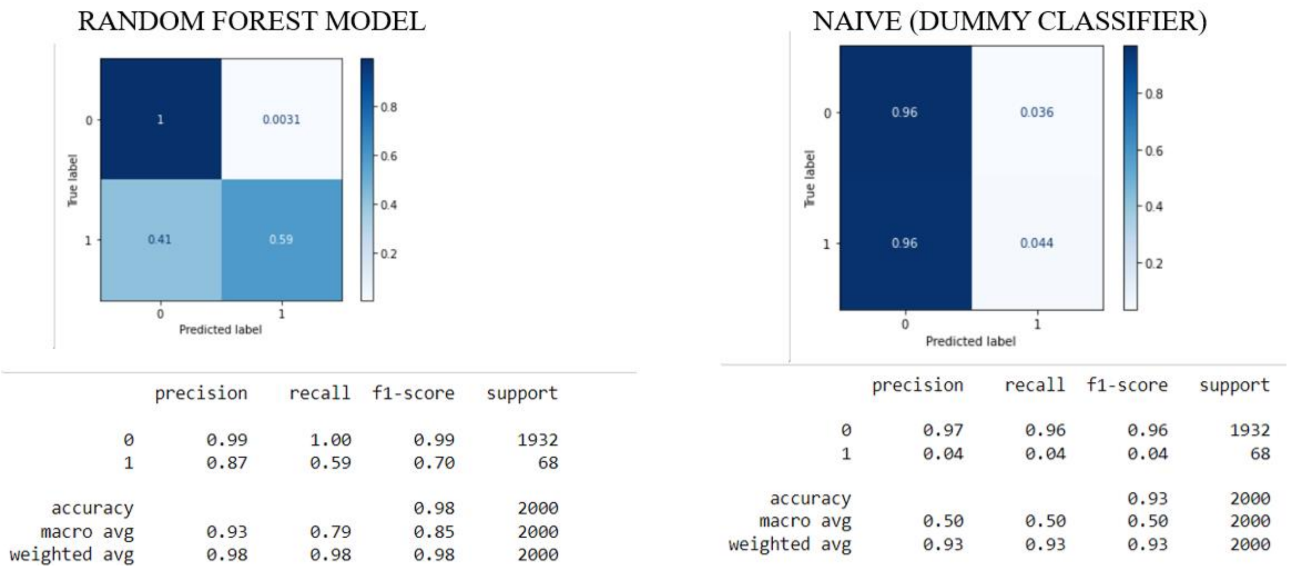
```
ai4i.drop("productid",axis=1,inplace=True)ai4i.
drop("TWF", axis=1, inplace=True)
ai4i.drop("HDF", axis=1,
inplace=True)ai4i.drop("PWF", axis=1,
inplace=True) ai4i.drop("OSF", axis=1,
inplace=True) ai4i.drop("RNF", axis=1,
inplace=True)
```

Now after performing all the data preprocessing techniques data is ready to be used for our interest where we can train data on different models to get our desired results

3. Results of single models

Random forest and Naive (Dummy classifier)

As we trained our data through random forest and naïve we found out that random forest is giving us an accuracy of 98% even though it looks great if we look at the truth table the model is only predicting machine failure correctly 59% of times and 100% correctly predicting when machine is not failure we mainly concerned about identifying machine failure so recall score becomes more important even in dummy classifier bulk of data is classified as machine not failure as shown in the figure



RANDOM FOREST MODEL

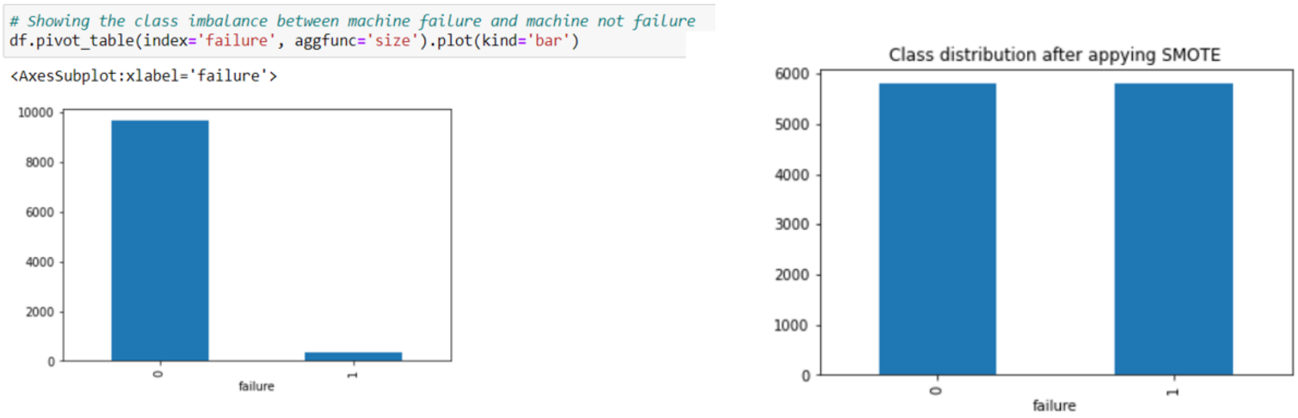
```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1932
           1       0.87      0.59      0.70        68

    accuracy                           0.98      2000
   macro avg       0.93      0.79      0.85      2000
weighted avg       0.98      0.98      0.98      2000
```

NAIVE (DUMMY CLASSIFIER)

```
              precision    recall  f1-score   support

           0       0.97      0.96      0.96      1932
           1       0.04      0.04      0.04        68

    accuracy                           0.93      2000
   macro avg       0.50      0.50      0.50      2000
weighted avg       0.93      0.93      0.93      2000
```

Fig3.1 truth table of random forest and dummy classifier

To over come this problem Smote technique have been used . a statistical method for evenly expanding the number of instances in your dataset is Synthetic Minority Oversampling Technique (SMOTE). The component creates new instances from minority situations that you specify as input that already exist. The quantity of majority cases remains unchanged as a result of this SMOTE implementation.



```
# Showing the class imbalance between machine failure and machine not failure
df.pivot_table(index='failure', aggfunc='size').plot(kind='bar')
```

```
<AxesSubplot:xlabel='failure'>
```

Fig3.2 dataset before and after SMOTE

As we can see in fig 3.2 our dataset contains many more machines that are not failures than those that are, therefore the dummy classifier predicts more machines as not failures than as failures. This explains why our Random Forest Classifier excelled at accurately recognizing machine failure rather than machine failure not failure. SMOTE increases recall at the cost of lower precision.

Once the data is balanced now we got Model structure selection based on ROC and AUC curve we trained our data set with four models as follows Logistic regression, K-NN, Decision Tree, and Random forest ROC and AUC curves are given below in fig 3.3

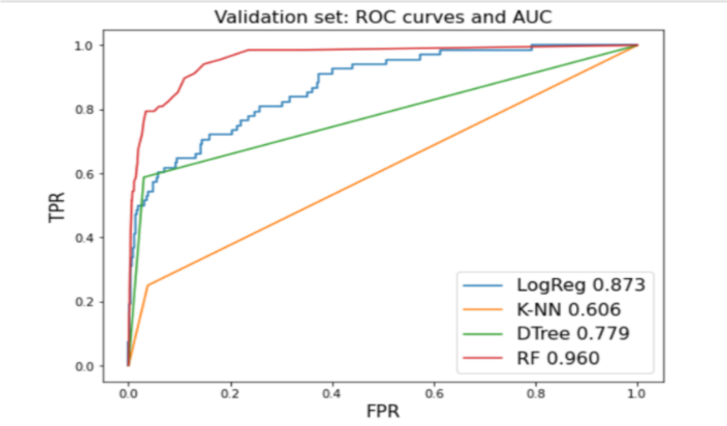


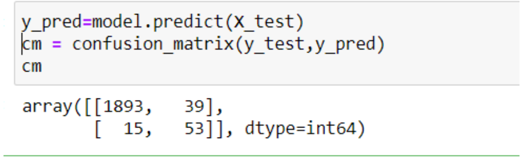
Fig 3.3 ROC and AUC

AUC is High for Random forest classifier with the value of 0.96.

From the comparison of overall results it turns out that the random forest is the best model after performing validation tests to all the models with good precision and recall

Models	Precision		Recall		F1-score		Accuracy
	0	1					
Logistic regression	0.99	0.16	0.86	.75	0.92	0.26	.85
KNN	0.98	0.33	0.93	.38	.95	.23	.91
Random forest	0.99	.51	.98	.63	.98	.56	.97
Decision tree	0.99	.33	.95	.71	.97	.45	.94

We again created the model for Random Forest Classifier utilizing those optimal hyperparameters that we obtained from Randomized Search CV. and after training and obtaining the confusion matrix we know that true positives(1893) and true negatives(53) values are much more higher than false positives(39) and false negatives(15) as we can seen from the given table 3.1



```
y_pred=model.predict(X_test)
cm = confusion_matrix(y_test,y_pred)
cm
```

```
array([[1893,   39],
       [  15,   53]], dtype=int64)
```

Table:3.1 confusion matrix

4. FIRST VARIABLE SELECTION:

Lasso:

To get the best variables Pipeline has been built with scaled data and LassoRegressor module with default value of alphas as 0.1. To find the best parameters used GridSearchCV and the best parameter we have is model_alpha as 0.1. As coefficients with less importance shrink to 0, we have 4 coefficients left with 1 which are considered as best parameters. Below are the best 4 variables we have:

Best features using Lasso: array(['air_temp', 'torque', 'Type_M'], dtype=object)
For a dataset with d features, if we apply the hit and trial method with all possible combinations of features then total (2^d – 1) models need to be evaluated for a significant set of features. It is a time-consuming approach, therefore, we use feature selection techniques to find out the smallest set of features more efficiently.

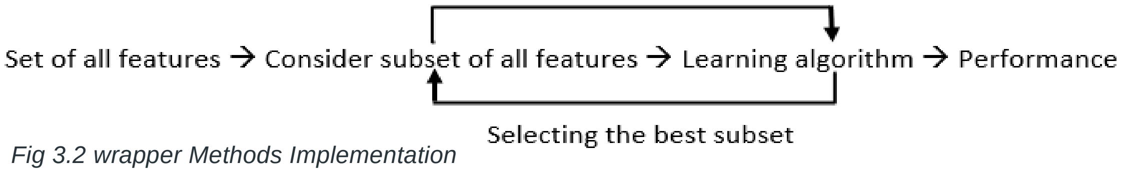


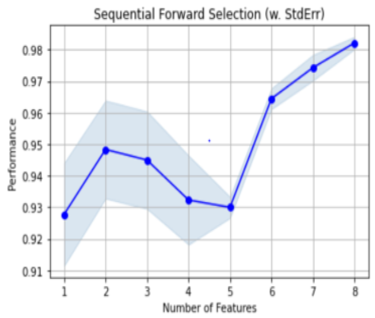
Fig 3.2 wrapper Methods Implementation

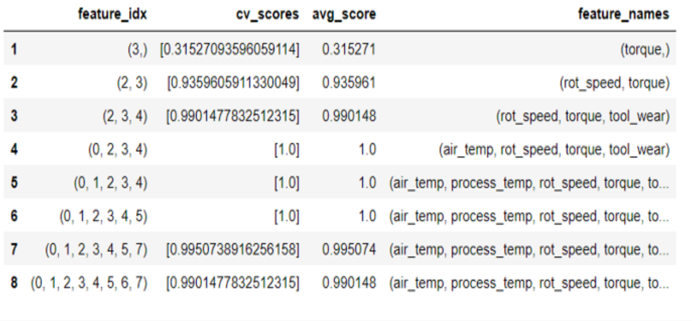
There are three types of feature selection techniques: Filter methods, Wrapper methods, and Embedded methods from which we used wrapper method. wrapper method employs a greedy search strategy by comparing every potential feature combination to the evaluation criterion. it chooses the feature combination that produces the best outcomes for the given machine learning method. The flow chart of wrapper method is given in figure 3.2

5. Bi-directional elimination

The most commonly used wrapper methods are bi-directional elimination(stepwise selection). It is a combination of forward selection and backward elimination. The results are given below in figures

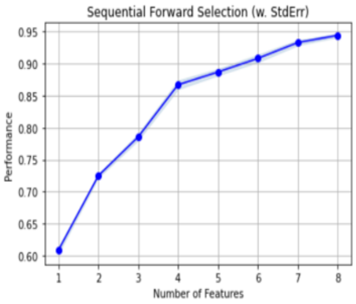
Part2 Best Selected Model(Random Forest):

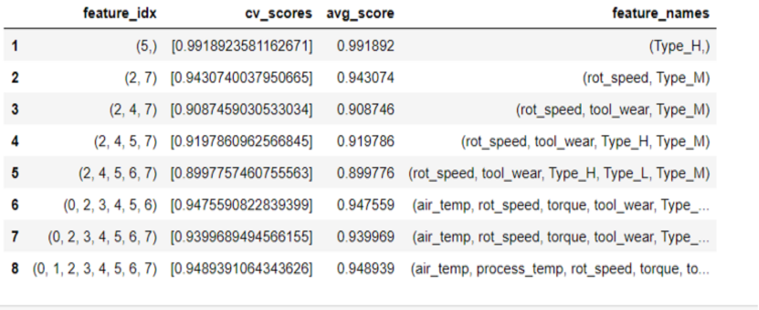




	feature_idx	cv_scores	avg_score	feature_names
1	(3,)	[0.31527093596059114]	0.315271	(torque,)
2	(2, 3)	[0.9359605911330049]	0.935961	(rot_speed, torque)
3	(2, 3, 4)	[0.9901477832512315]	0.990148	(rot_speed, torque, tool_wear)
4	(0, 2, 3, 4)	[1.0]	1.0	(air_temp, rot_speed, torque, tool_wear)
5	(0, 1, 2, 3, 4)	[1.0]	1.0	(air_temp, process_temp, rot_speed, torque, to...
6	(0, 1, 2, 3, 4, 5)	[1.0]	1.0	(air_temp, process_temp, rot_speed, torque, to...
7	(0, 1, 2, 3, 4, 5, 7)	[0.9950738916256158]	0.995074	(air_temp, process_temp, rot_speed, torque, to...
8	(0, 1, 2, 3, 4, 5, 6, 7)	[0.9901477832512315]	0.990148	(air_temp, process_temp, rot_speed, torque, to...

Part3 Best Selected Model(Non-Linear SVM):



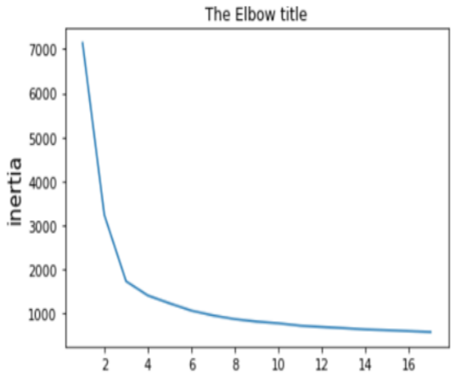


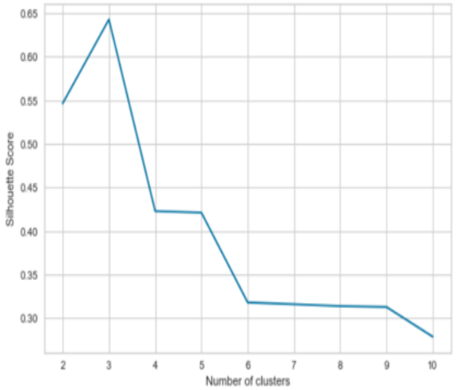
	feature_idx	cv_scores	avg_score	feature_names
1	(5,)	[0.9918923581162671]	0.991892	(Type_H,)
2	(2, 7)	[0.9430740037950665]	0.943074	(rot_speed, Type_M)
3	(2, 4, 7)	[0.9087459030533034]	0.908746	(rot_speed, tool_wear, Type_M)
4	(2, 4, 5, 7)	[0.9197860962566845]	0.919786	(rot_speed, tool_wear, Type_H, Type_M)
5	(2, 4, 5, 6, 7)	[0.8997757460755563]	0.899776	(rot_speed, tool_wear, Type_H, Type_L, Type_M)
6	(0, 2, 3, 4, 5, 6)	[0.9475590822839399]	0.947559	(air_temp, rot_speed, torque, tool_wear, Type_...
7	(0, 2, 3, 4, 5, 6, 7)	[0.9399689494566155]	0.939969	(air_temp, rot_speed, torque, tool_wear, Type_...
8	(0, 1, 2, 3, 4, 5, 6, 7)	[0.9489391064343626]	0.948939	(air_temp, process_temp, rot_speed, torque, to...

- For both the methods, high performance is when we have all the features

6. Clustering

clustering is an unsupervised learning technique to extract natural groupings or labels from predefined classes and prior information. This is an important technique to use for Exploratory Data Analysis (EDA) to discover hidden groupings from data. Usually, It is clustering is used to discover insights regarding data distributions and feature engineering to generate a new class for other algorithms the results after performing clustering is given below



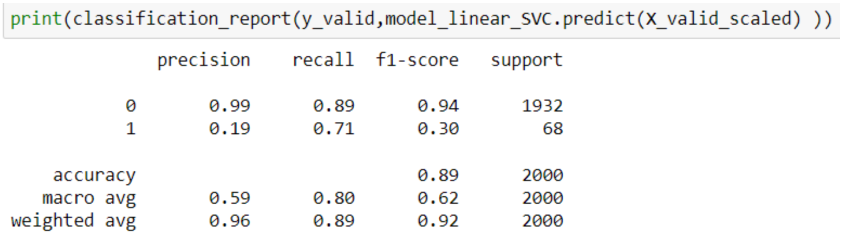


For clusters=3 there is less inertia and high silhouette score

Clustering really helps in identifying new insight of the features which helps in prediction

SVM

The Support Vector Machine is a supervised learning technique that is primarily used for classification. Although it can be used for regression, The key concept is that the algorithm searches for the best hyperplane that may be used to categorize new data points based on the labeled data (training data). The hyperplane is a straight line in two dimensions.

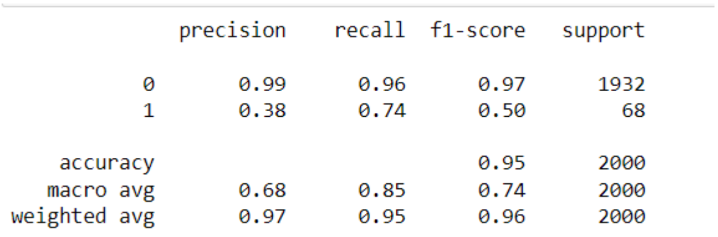


```
print(classification_report(y_valid,model_linear_SVC.predict(X_valid_scaled) ))

              precision    recall  f1-score   support

           0       0.99      0.89      0.94      1932
           1       0.19      0.71      0.30        68

    accuracy                           0.89      2000
   macro avg       0.59      0.80      0.62      2000
weighted avg       0.96      0.89      0.92      2000
```

Again using randomized search cv we have found the best hyper parameters this is the classification for that hyper parameter using SVM with linear kernel the result is as shown in figure 5.1

Fig 5.1 result of SVM with linear kernel

SVM with Non-Linear kernel (kernel=poly)



```
              precision    recall  f1-score   support

           0       0.99      0.96      0.97      1932
           1       0.38      0.74      0.50        68

    accuracy                           0.95      2000
   macro avg       0.68      0.85      0.74      2000
weighted avg       0.97      0.95      0.96      2000
```

table on the left shows SVM with non-linear kernel which has accuracy of 95

Compared to non-linear SVM, linear SVM is less prone to overfitting. And you need to decide which kernel to use based on your situation: if your feature number is really small compared to the training sample, use a linear kernel; if your feature number is small but the training sample is large, you may also need a linear kernel but try to add more features.

7. Data visualization

Data visualization is one of the most crucial components of dimensionality reduction. By reducing the dimensionality of the data to two or three, it is feasible to depict the data on a two-dimensional or three-dimensional plot, allowing for the analysis of these patterns in terms of clusters and other concepts.it helps in any prediction model

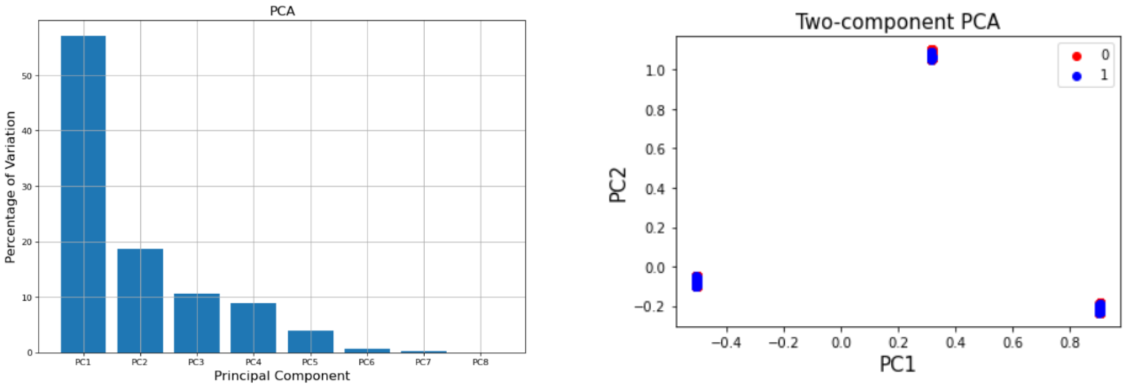


Fig6.1 For linear dimensionality reduction

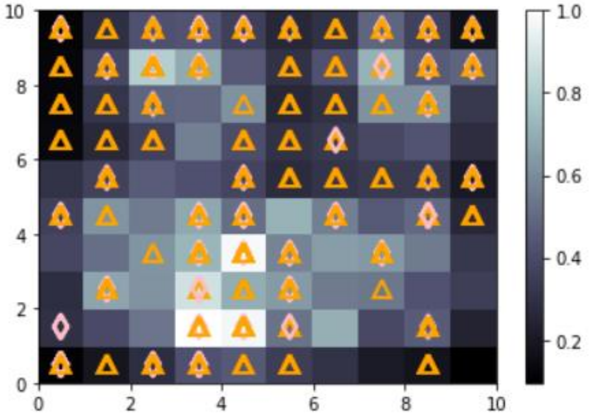
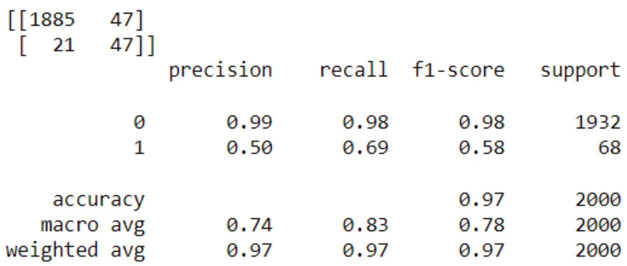


Fig6.2 For nonlinear dimensionality reduction

8. Ensemble modeling

Using a variety of modeling algorithms or training data sets, ensemble modeling is the process of building numerous varied models to predict an outcome. The ensemble model then combines each base model's forecast into a single overall prediction for the unobserved data.

The goal of employing ensemble models is to lower the prediction's generalization error. When using the ensemble approach, the prediction error of the model lowers as long as the basic models are diverse and independent.



```
[[1885   47]
 [  21   47]]
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1932
           1       0.50      0.69      0.58        68

    accuracy                           0.97      2000
   macro avg       0.74      0.83      0.78      2000
weighted avg       0.97      0.97      0.97      2000
```

Tab 7.1Classification report of Ensemble

The ensemble model behaves and functions as a single model even when it has numerous basis models. The majority of real-world data mining solutions use ensemble modeling methods.

The classification report of the ensemble model is shown in table7.1

9.Discussions &Conclusion

For this data set, it will be a serious issue if actual machine is a failure but test shows machine is not failure. So, recall score is very important We find Non-Linear SVM has the best model for our dataset. Using bidirectional feature elimination method, we found all the features has equal importance in maintaining good recall score. We had a good recall score for both positive and negative classes.

11.References

1. Feature Selection using Wrapper Method - Python Implementation (analyticsvidhya.com)
2. SVM and Kernel SVM. Learn about SVM or Support Vector… | by Czako Zoltan | Towards Data Science
3. Major Kernel Functions in Support Vector Machine (SVM) - GeeksforGeeks
4. The 5 Clustering Algorithms Data Scientists Need to Know | by George Seif | Towards Data Science
5. Ensemble Modeling - an overview | ScienceDirect Topics