AME : Appraisal Matrix for Employees

Shubham Saha - 16155014
Department of Information Technology
University of Information Technology & Sciences
shuvsaha7@gmail.com

Abstract

Now we are living in the era of Information Technology. Everything around us is having a touch of Information Technology. Basically what it is doing is making our daily life easier then previous. Almost every sector is trying to have a touch of automated systems using IT. But there is one sector of work where automated work haven't focused, which is on every company or organizations HR department. Every year or some sort of time companies or organizations need to give appraisal to their employees. For this they haven't any automated system to predict who is eligible for appraisal. In this paper I will introduce AME, an automated system for predicting employee's appraisal. This is basically a proposed work which will be implemented soon with a good hands on it.

Keywords

I. Introduction

In companies or organizations HR department works a lot at a time of appraisal. Because it is a hard task to identify which employee is well suited for the appraisal. All things are done by hands-on the department head officers. Sometimes unfair decision is been taken by them, because by their some mistake a eligible employee misses his/her appraisal. This may lead to a loss of that company or organization when that employee leaves from his/her work. To solve this problem the only way is to use an automated system which will predict every employee's appraisal automatically.

For this AME is introduced. What AME will do is minimize the work load of HR department so that no more unfair may occurs and they can save their valuable time on doing other important things for their working zone.

A. What is AME ?

AME is an automated system which is build to minimize the work load of HR department. Basically AME is a machine learning based system which is trained with a dataset and based on that dataset the system will predict that a particular employee is eligible for appraisal or not by feeding it that employee's performance list. Dataset can be altered based on a company or organization's policy. As this is a proposed work so for this I have created a dataset based on any company or organizations basic policy. AME is build using machine learning algorithm named Support Vector Machine algorithm, which is categorized as a Supervised learning's Classification Analysis.

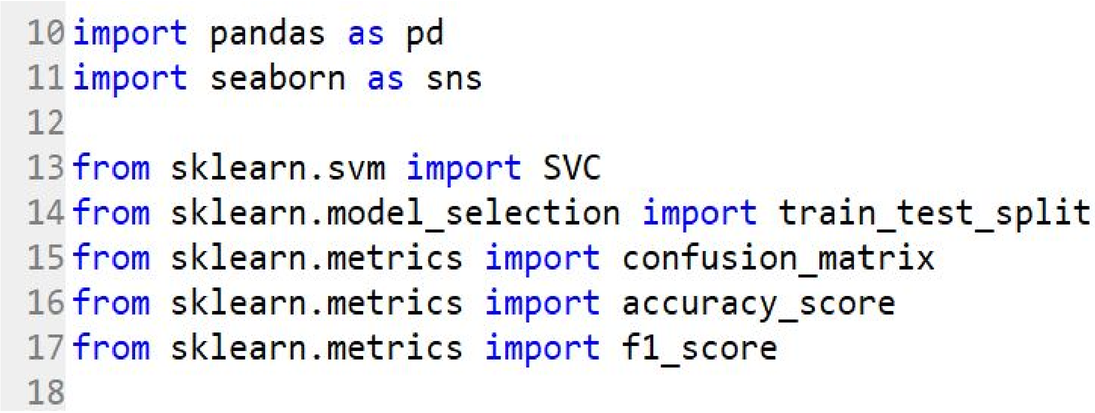
B. What is Support Vector Machine algorithm ?

SVM stands for Support Vector Machine. It is a machine learning approach used for classification and regression analysis. It depends on supervised learning models and trained by learning algorithms. They analyze the large amount of data to identify patterns from them. An SVM generates parallel partitions by generating two parallel lines. For each category of data in a high-dimensional space and uses almost all attributes. It separates the space in a single pass to generate flat and linear partitions. Divide the 2 categories by a clear gap that should be as wide as possible. Do this partitioning by a plane called *hyperplane.* An SVM creates hyperplanes that have the largest margin in a high-dimensional space to separate given data into classes. The margin between the 2 classes represents the longest distance between closest data points of those classes. The larger the margin, the lower is the generalization error of the classifier.

II. Working procedure & Implementation of AME

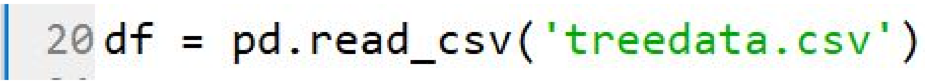
As it is mentioned before that AME is a proposed work. But the prototype which is now working is build with machine learning's SVM algorithm. This is implemented in Python Programming Language. Here the dataset which I manually created is trained in the SVM and after that some value to predict will be added. Thus our predicted result will be shown. Below every steps of AME the implement process is described. I have implemented this using Anaconda Navigator's Spyder environment.

A. **Step 1: Importing necessary packages**



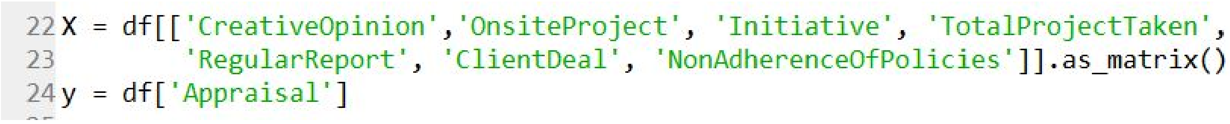
```
import pandas as pd
import seaborn as sns

from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score

```

B. **Step 2: Importing dataset**



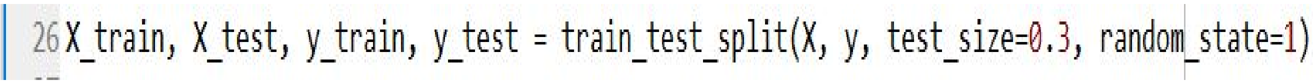
```
df = pd.read_csv('treedata.csv')
```

C. **Step 3: Separate working data and class data**



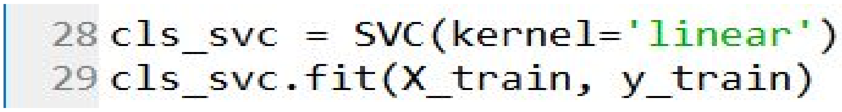
```
X = df[['CreativeOpinion','OnsiteProject', 'Initiative', 'TotalProjectTaken',
         'RegularReport', 'ClientDeal', 'NonAdherenceOfPolicies']].as_matrix()
y = df['Appraisal']
```

D. **Step 4: Split test and train data**



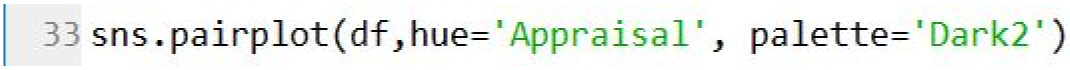
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

E. **Step 5: Train SVM with test data**



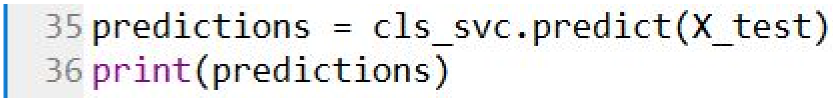
```
cls_svc = SVC(kernel='linear')
cls_svc.fit(X_train, y_train)
```

F. **Step 6: Plotting those data using seaborn**



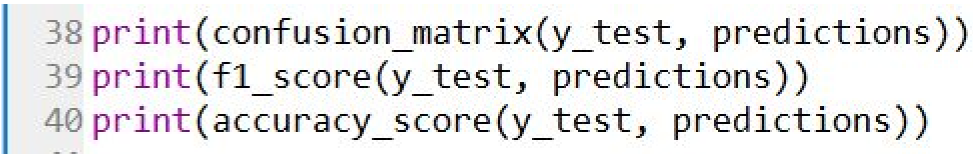
```python
sns.pairplot(df,hue='Appraisal', palette='Dark2')
```

G. **Step 7: Testing the model with test data and printing it**



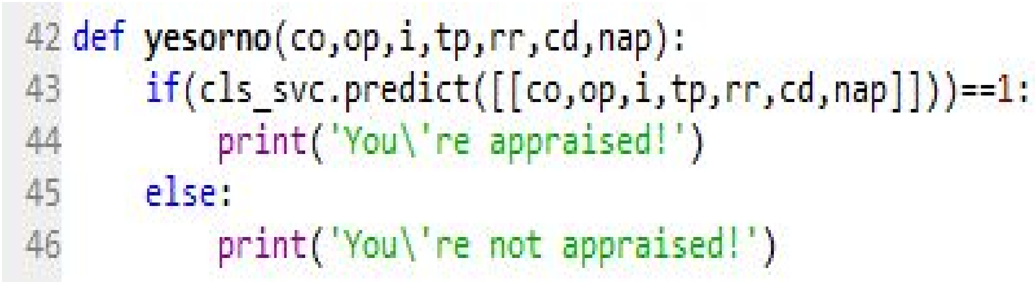
```python
predictions = cls_svc.predict(X_test)
print(predictions)
```

H. **Step 8: Evaluating the model from Confusion matrix, f1_score & accuracy_score**



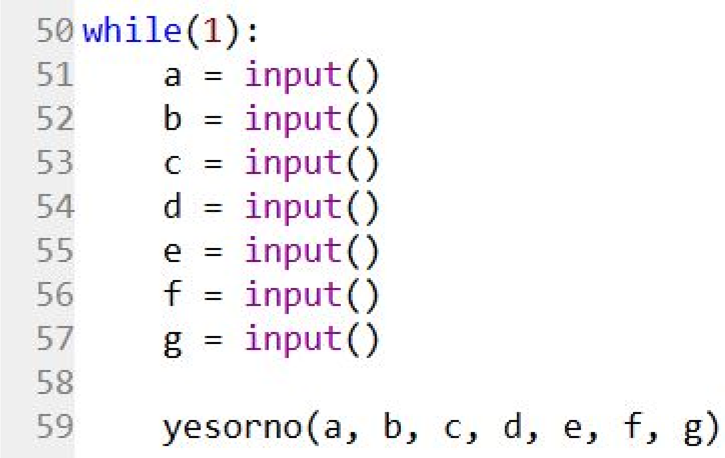
```python
print(confusion_matrix(y_test, predictions))
print(f1_score(y_test, predictions))
print(accuracy_score(y_test, predictions))
```

I. **A function that will predict the results by user data**



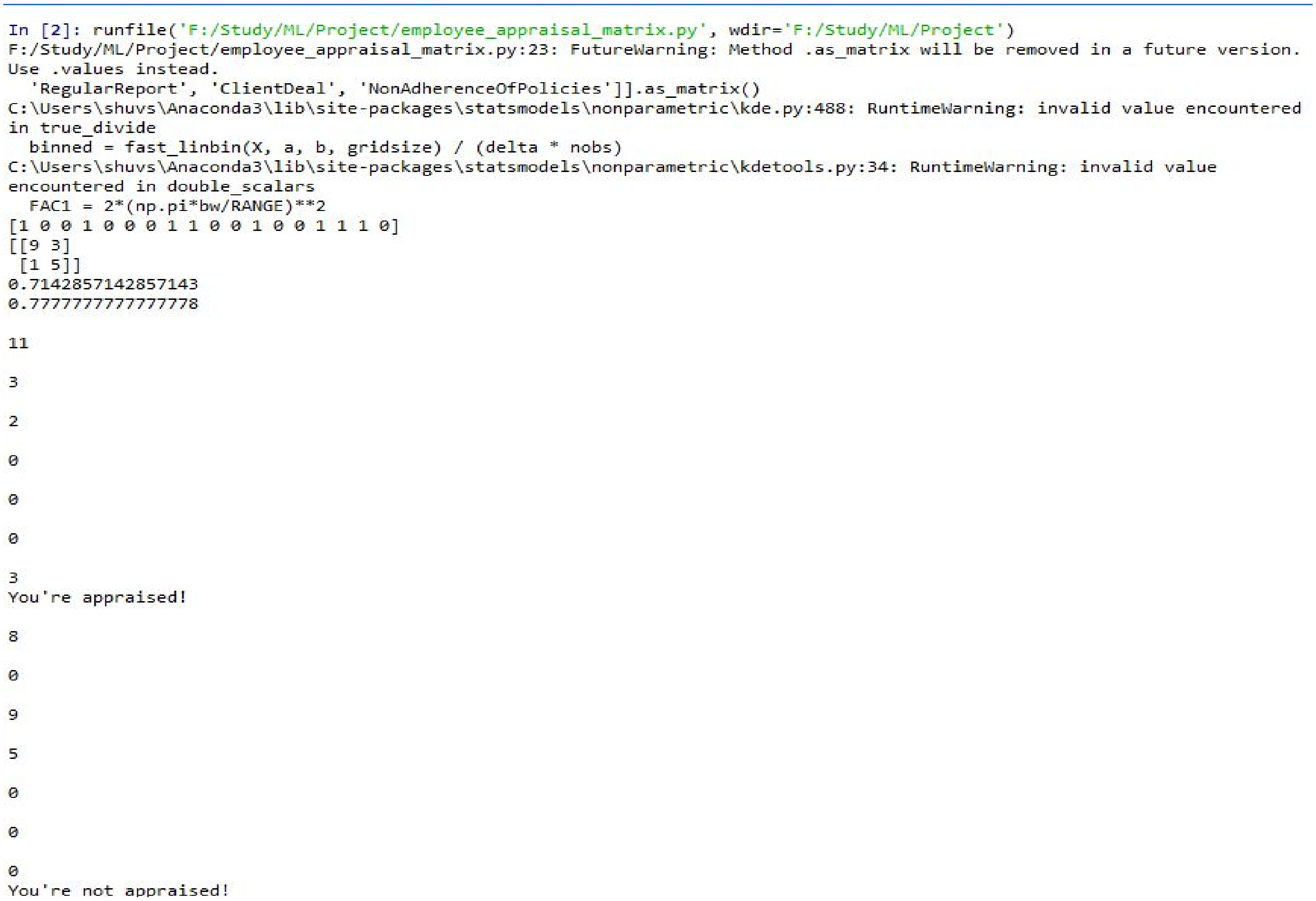
```python
def yesorno(co,op,i,tp,rr,cd,nap):
    if(cls_svc.predict([[co,op,i,tp,rr,cd,nap]]))==1:
        print('You\'re appraised!')
    else:
        print('You\'re not appraised!')
```

J. **Infinite loop that will take user input and show results by using the yesorno function**



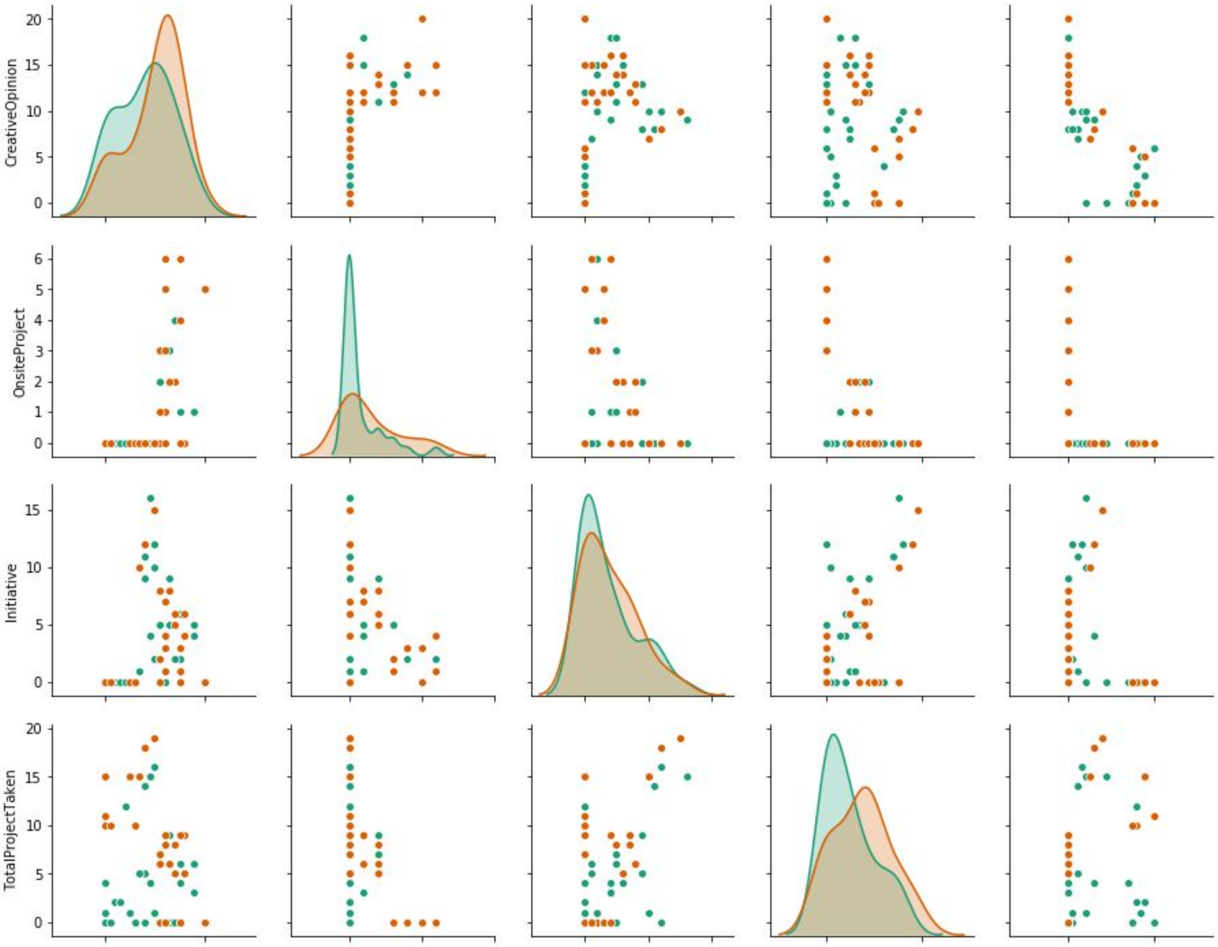
```
while(1):
    a = input()
    b = input()
    c = input()
    d = input()
    e = input()
    f = input()
    g = input()

    yesorno(a, b, c, d, e, f, g)
```

K. **Sample Output**



```
In [2]: runfile('F:/Study/ML/Project/employee_appraisal_matrix.py', wdir='F:/Study/ML/Project')
F:/Study/ML/Project/employee_appraisal_matrix.py:23: FutureWarning: Method .as_matrix will be removed in a future version.
Use .values instead.
  'RegularReport', 'ClientDeal', 'NonAdherenceOfPolicies']].as_matrix()
C:\Users\shuvs\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py:488: RuntimeWarning: invalid value encountered
in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\Users\shuvs\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetools.py:34: RuntimeWarning: invalid value
encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
[1 0 0 1 0 0 0 1 1 0 0 1 0 0 1 1 1 0]
[[9 3]
 [1 5]]
0.7142857142857143
0.7777777777777778

11

3

2

0

0

0

3
You're appraised!

8

0

9

5

0

0

0
You're not appraised!
```

L. Sample Plotting Output



III. Future Work

As i mentioned earlier that this work is just a proposed project so in future I will try to make it more efficient and user friendly by creating a GUI for this system and also add some features like how much appraisal will an employee get. That will also be featured by plotting them with chart. And some more features will be added based on a company or organizations policy.

IV. Conclusion

AME is an automated system to minimize the working load of HR department and also to reduce the unfair decision to employees about their appraisal. Any company or organization can use this automated system for their better use in their company or organization. This system is scalable and for that any type of changes based on certain policy can be altered in making.

V. References

1.https://data-flair.training/blogs/svm-support-vector-machine-tutorial/

2.https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

3.http://www.statsoft.com/textbook/support-vector-machines

4.https://www.youtube.com/watch?v=TtKF996oEl8&t=965s

5.https://www.youtube.com/watch?v=g8D5YL6cOSE&t=1144s

6.https://github.com/adashofdata/muffin-cupcake