Parkinson Disease | Downloads/ | PARKINSON_DISEASE_PREDICTI | localhost:8888/notebooks/Downloads/PARKINSON_DISEASE_PREDICTION.ipynb

HighQStudy

jupyter PARKINSON_DISEASE_PREDICTION Last Checkpoint: Last Wednesday at 10:32 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run Markdown

Model Building

In [148]:
```
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.datasets import make_classification
from sklearn.metrics import plot_confusion_matrix
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

1.Random Forest Classifier

In [149]:
```
rfc = RandomForestClassifier()
rfc.fit(x_train, y_train)
predRF = rfc.predict(x_test)
print(classification_report(y_test, predRF))
```

```
              precision    recall  f1-score   support

           0       0.89      0.97      0.93        33
           1       0.96      0.85      0.90        26

    accuracy                           0.92        59
   macro avg       0.92      0.91      0.91        59
weighted avg       0.92      0.92      0.91        59
```

Type here to search ENG 12:04 15-11-2022

Parkinson Disease | Downloads/ | PARKINSON_DISEASE_PREDICTI | localhost:8888/notebooks/Downloads/PARKINSON_DISEASE_PREDICTION.ipynb

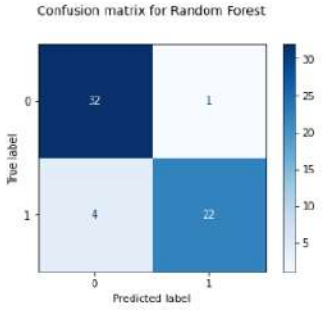
HighQStudy

jupyter PARKINSON_DISEASE_PREDICTION Last Checkpoint: Last Wednesday at 10:32 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run Markdown

In [150]:
```
plot_confusion_matrix(rfc, x_test, y_test, cmap=plt.cm.Blues)
plt.title('Confusion matrix for Random Forest', y=1.1)
plt.show()
```



2.SVM

SVM With Linear Kernel

In [151]:
```
model = svm.SVC(kernel='linear')
# Training the SVM model with training data
model.fit(x_train, y_train)
```

Type here to search ENG 12:04 15-11-2022

Parkinson Disease | Downloads/ | PARKINSON_DISEASE_PREDICTIC

localhost:8888/notebooks/Downloads/PARKINSON_DISEASE_PREDICTION.ipynb

HighQStudy

jupyter PARKINSON_DISEASE_PREDICTION Last Checkpoint: Last Wednesday at 10:32 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Markdown

2.SVM

SVM With Linear Kernel

In [151]:
```
model = svm.SVC(kernel='linear')
# Training the SVM model with training data
model.fit(x_train, y_train)
```

Out[151]:
```
SVC(kernel='linear')
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [152]:
```
predsvm = model.predict(x_test)
print(classification_report(y_test, predsvm))
```

```
              precision    recall  f1-score   support

           0       0.76      0.97      0.85        33
           1       0.94      0.62      0.74        26

    accuracy                           0.81        59
   macro avg       0.85      0.79      0.80        59
weighted avg       0.84      0.81      0.81        59
```

In [153]:
```
plot_confusion_matrix(model, x_test, y_test, cmap=plt.cm.Blues)
plt.title('Confusion matrix for Random Forest', y=1.1)
plt.show()
```

Confusion matrix for Random Forest

Type here to search ENG 12:04 15-11-2022

Parkinson Disease | Downloads/ | PARKINSON_DISEASE_PREDICTI | +

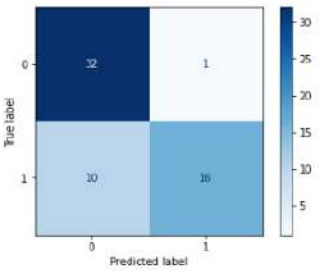
localhost:8888/notebooks/Downloads/PARKINSON_DISEASE_PREDICTION.ipynb

HighQStudy

jupyter PARKINSON_DISEASE_PREDICTION Last Checkpoint: Last Wednesday at 10:32 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run Markdown



3.Decision Tree Classifier

In [154]:
```
dtc = DecisionTreeClassifier()
dtc.fit(x_train, y_train)
predDT = dtc.predict(x_test)
print(classification_report(y_test, predDT))
```

```
              precision    recall  f1-score   support

           0       0.97      0.88      0.92        33
           1       0.86      0.96      0.91        26

    accuracy                           0.92        59
   macro avg       0.91      0.92      0.91        59
weighted avg       0.92      0.92      0.92        59
```

Type here to search

ENG 12:04 15-11-2022

Parkinson Disease | Downloads/ | PARKINSON_DISEASE_PREDICTI

localhost:8888/notebooks/Downloads/PARKINSON_DISEASE_PREDICTION.ipynb

HighQStudy

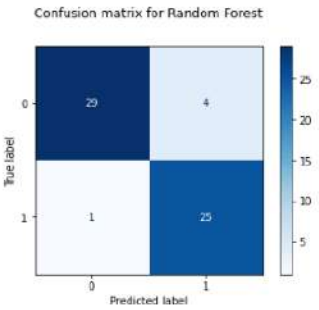
jupyter PARKINSON_DISEASE_PREDICTION Last Checkpoint: Last Wednesday at 10:32 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Markdown

In [155]:
```
plot_confusion_matrix(dtc, x_test, y_test, cmap=plt.cm.Blues)
plt.title('Confusion matrix for Random Forest', y=1.1)
plt.show()
```



Comparision Table

In [156]:
```
from sklearn.metrics import precision_score,recall_score ,accuracy_score, f1_score, r2_score, log_loss

chart = {
        'Metric':["Accuracy", "F1-Score", "Recall", "Precision", "R2-Score"],
        'DT':[accuracy_score(y_test, predDT), f1_score(y_test, predDT), recall_score(y_test, predDT), precision_score(y_test, pre
        'RF':[accuracy_score(y_test, predRF), f1_score(y_test, predRF), recall_score(y_test, predRF), precision_score(y_test, pre
        'SVM':[accuracy_score(y_test, predsvm), f1_score(y_test, predsvm), recall_score(y_test, predsvm), precision_score(y_test,
        }
```

Type here to search

ENG 12:05 15-11-2022

Parkinson Disease | Downloads/ | PARKINSON_DISEASE_PREDICTI | localhost:8888/notebooks/Downloads/PARKINSON_DISEASE_PREDICTION.ipynb

HighQStudy

jupyter PARKINSON_DISEASE_PREDICTION Last Checkpoint: Last Wednesday at 10:32 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Markdown

Comparision Table

In [156]:
```
from sklearn.metrics import precision_score,recall_score ,accuracy_score, f1_score, r2_score, log_loss

chart = {
        'Metric':["Accuracy", "F1-Score", "Recall", "Precision", "R2-Score"],
        'DT':[accuracy_score(y_test, predDT), f1_score(y_test, predDT), recall_score(y_test, predDT), precision_score(y_test, pre
        'RF':[accuracy_score(y_test, predRF), f1_score(y_test, predRF), recall_score(y_test, predRF), precision_score(y_test, pre
        'SVM':[accuracy_score(y_test, predsvm), f1_score(y_test, predsvm), recall_score(y_test, predsvm), precision_score(y_test,
        }
chart = pd.DataFrame(chart)
```

In [157]:
```
display(chart)
```

	Metric	DT	RF	SVM
0	Accuracy	0.915254	0.915254	0.813559
1	F1-Score	0.909091	0.897959	0.744186
2	Recall	0.961538	0.846154	0.615385
3	Precision	0.862069	0.956522	0.941176
4	R2-Score	0.656177	0.656177	0.243590

USER INPUT TESTING

In [158]:
```
a=parkinsons_data.iloc[0].to_dict()              #  parkinson disease-----1
b=parkinsons_data.iloc[1].to_dict()              #  parkinson disease-----1
```

Type here to search ENG 12:05 15-11-2022