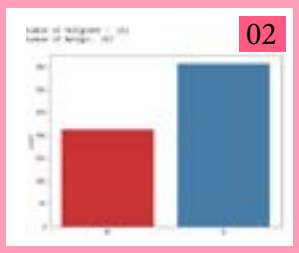
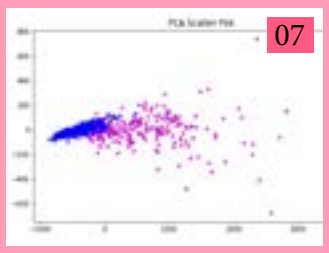
Pepgra

Breast Cancer Prediction

HIGHLIGHTS



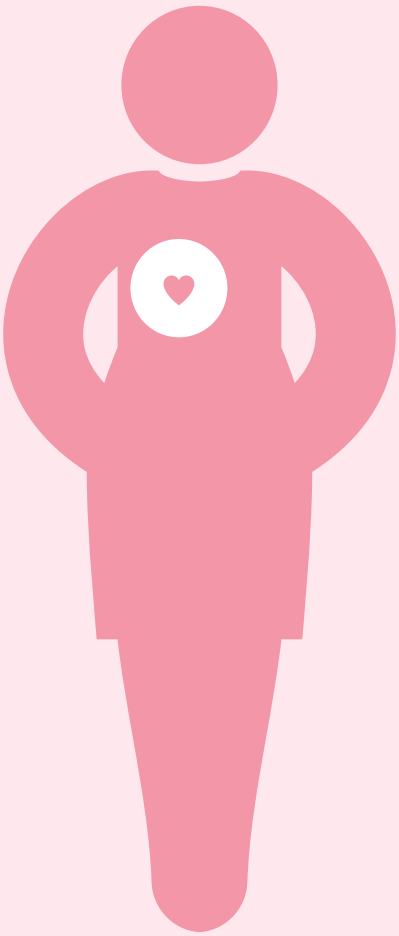
DATASET



Visualization

Predictive modeling

Pepgra
Drives Innovation



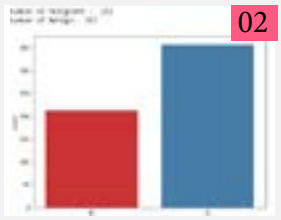
PREFACE

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society.

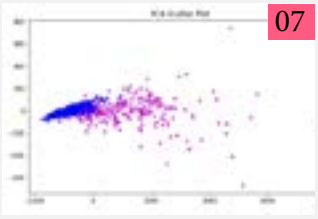
Objective

The intention of this study is to design a prediction system that can predict the incidence of the breast cancer at early stage by analysing smallest set of attributes that has been selected from the clinical dataset.

HIGHLIGHTS



DATASET



Visualization

Predictive modeling

Dataset

We have used Wisconsin breast cancer dataset (WBCD). The potential of the proposed method is obtained using classification accuracy which was obtained by comparing actual to predicted values.
Ten real-valued features:

- Radius (mean of distances from center to points on the perimeter)
- Texture (standard deviation of gray-scale values)
- Perimeter
- Area
- Smoothness (local variation in radius lengths)
- Compactness (perimeter^2 / area — 1.0)
- Concavity (severity of concave portions of the contour)
- Concave points (number of concave portions of the contour)
- Symmetry
- Fractal dimension (“coastline approximation” — 1)

For each feature there is mean, standard error and “worst” or largest which will be used for modelling.

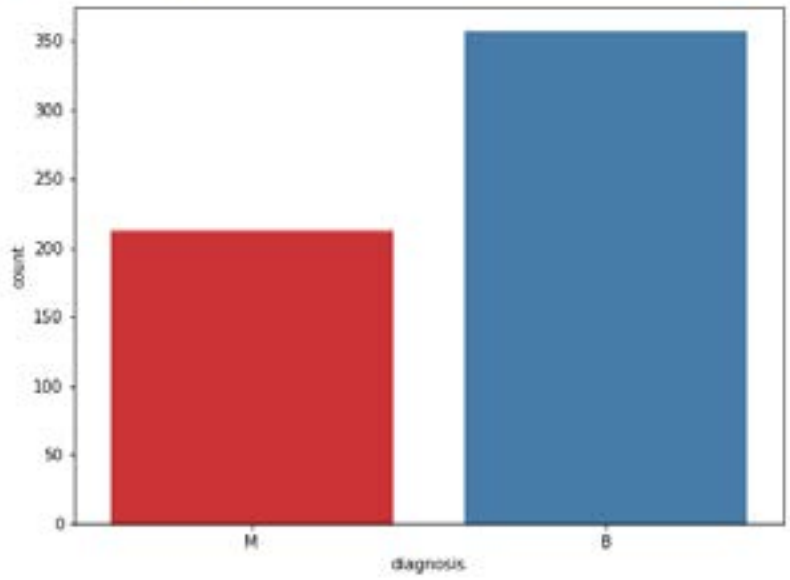
Shape of this dataset is (569, 33), in those 33 features we will see which should be taken for modelling.
Diagnosis is the target variable of our Dataset, which has two classes (M = malignant, B = benign).

> “Number of patients - 569 - 569.

```
data.diagnosis.unique()
```

```
array(['M', 'B'], dtype=object)
```

```
Number of Malignant :  212
Number of Benign:  357
```



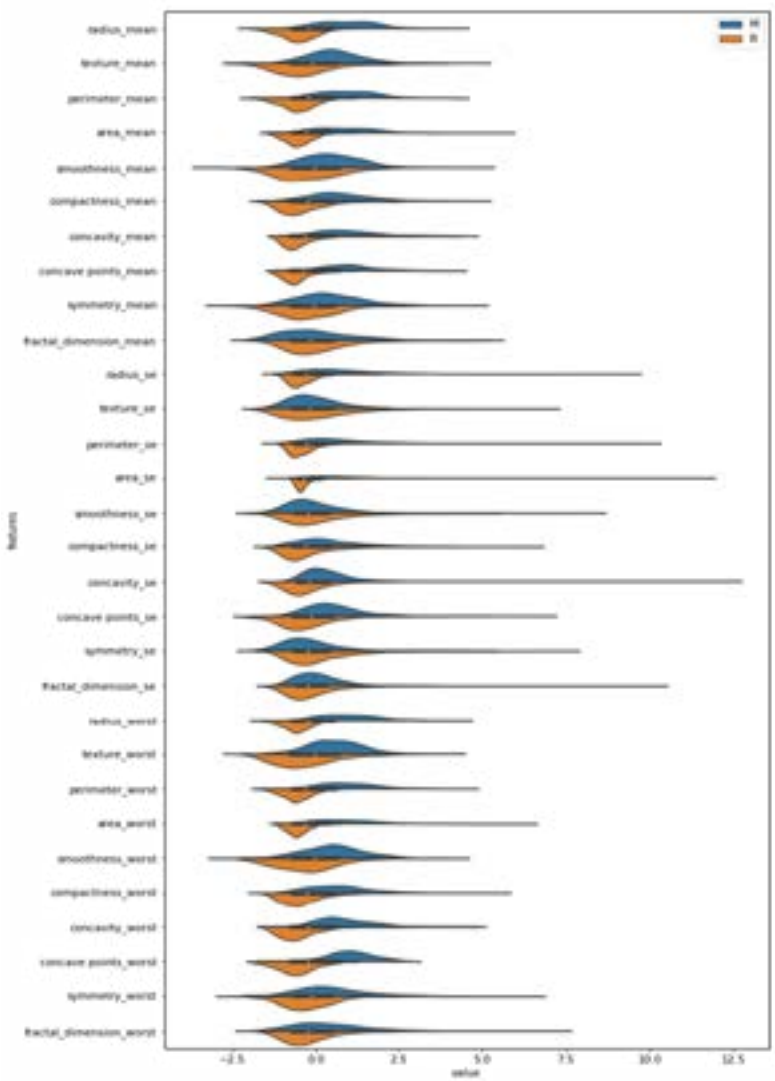
Standardisation

> **62%** of breast cancer cases are diagnosed at a localized stage, for which the 5-year survival rate is **99%**.

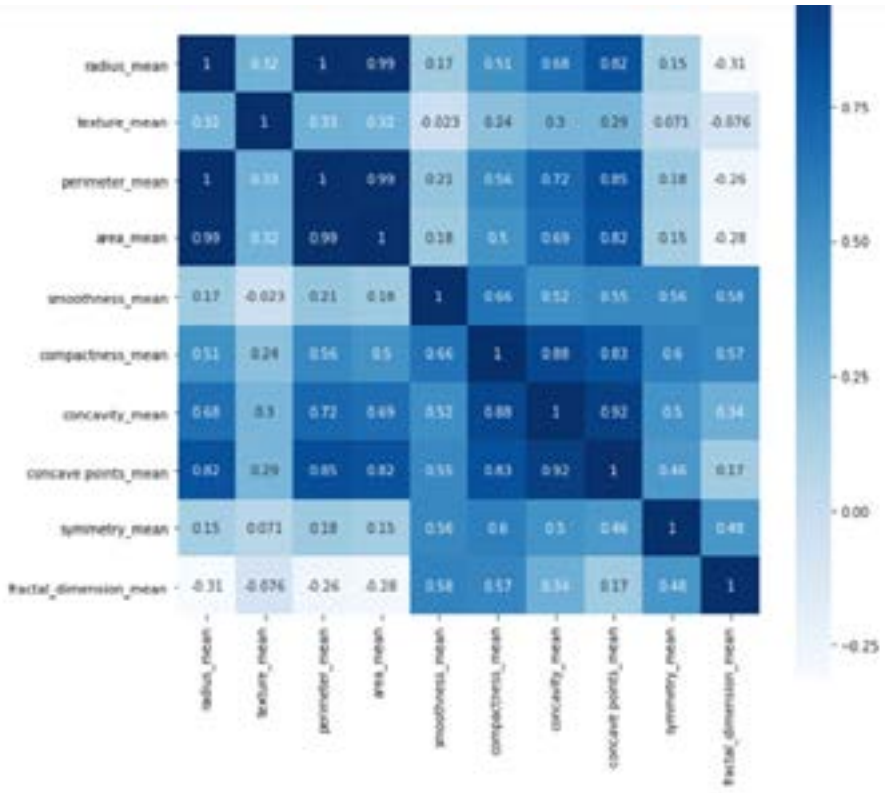
This is an important step before modelling as the features are needed to be in particular range. Consider this example, if feature radius_mean - 50 and smoothness_mean - 0.5 is fed into ML algorithm the machine thinks that age is more important as it has higher value. For this reason we need to normalize the dataset leaving out the target column.

Below is the code and plot of a violin plot from Seaborn for all the features and their values after standardisation.

```
plt.figure(figsize=(10,20))
sns.violinplot(x="value", y="features", hue="diagnosis", data=data_norm,split=True)
plt.legend(loc='best');
```



Now let us see the Correlation between each variables of Mean features from the dataset to analyse our data well. We will be using Heatmap from Seaborn package.



Principal Component

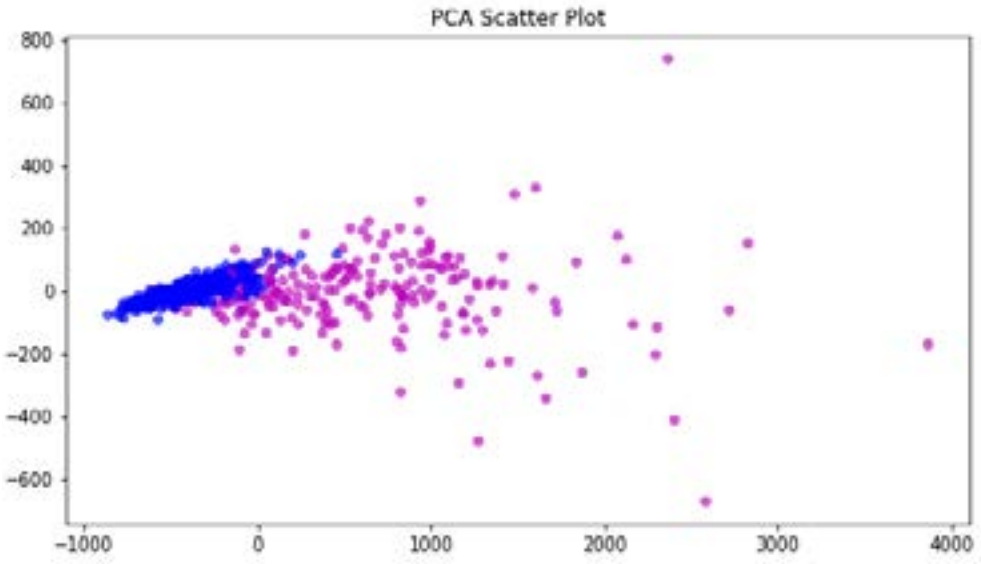
PCA is essentially a method that reduces the dimension of the feature space in such a way that new variables are orthogonal to each other (i.e. they are independent or not correlated).

Anyway, from the cancer data-set we see that it has 30 features, so let's reduce it to only 2 principal features and then we can visualize the scatter plot of these new independent variables.

Since the PCA components are orthogonal to each other and they are not correlated, we can expect to see malignant and benign classes as distinct

The two classes are well separated with the 2 principal components as new features. As good as it seems ,there might be some difficulty to linearly separate using simple algorithms.

> **PCA** is an unsupervised statistical technique used to examine the interrelations among a set of variables



> **SVM KERNAL USED ARE** - Linear and Radial basic function Kernel

PREDICTIVE MODELLING

We will be using different machine learning algorithms such as SVM(Support Vector Machine), Logistic Regression , Random forest classifier.

SVM

We will be using support vector machine from sklearn python package. There are two kernels in svm namely Linear and rbf , we will be testing both the algorithms.

```
Accuracy for SVM kernel= rbf is 0.631578947368421
Confusion Matrix
 [[108  63]
 [  0   0]]
Accuracy for SVM kernel= linear is 0.9649122807017544
Confusion Matrix
 [[106   4]
 [  2  59]]
```

Here we can see that rbf kernel is performing so poor and Linear kernel is better than it with accuracy of 96%.

Logistic Regression

This algorithm is also from sklearn python package, here the main hyperparameters are C and Penalty(L1 or L2)

```
The accuracy of the Logistic Regression is 0.9649122807017544
Confusion Matrix
 [[106   4]
 [  2  59]]
```

Random Forest

This algorithm is a type of ensemble model which works well for classification problems. Let us first find the important features and then predict the diagnosis.

> **RANDOM FORESTS** are an ensemble learning method for classification.

```
concave points_worst      0.150736
perimeter_worst           0.134192
concave points_mean       0.104082
radius_worst              0.089342
concavity_mean            0.084554
area_worst                0.070636
perimeter_mean            0.061195
concavity_worst           0.036497
area_mean                 0.035555
area_se                   0.032479
radius_mean               0.025707
radius_se                 0.024580
texture_worst             0.019092
texture_mean              0.018864
```

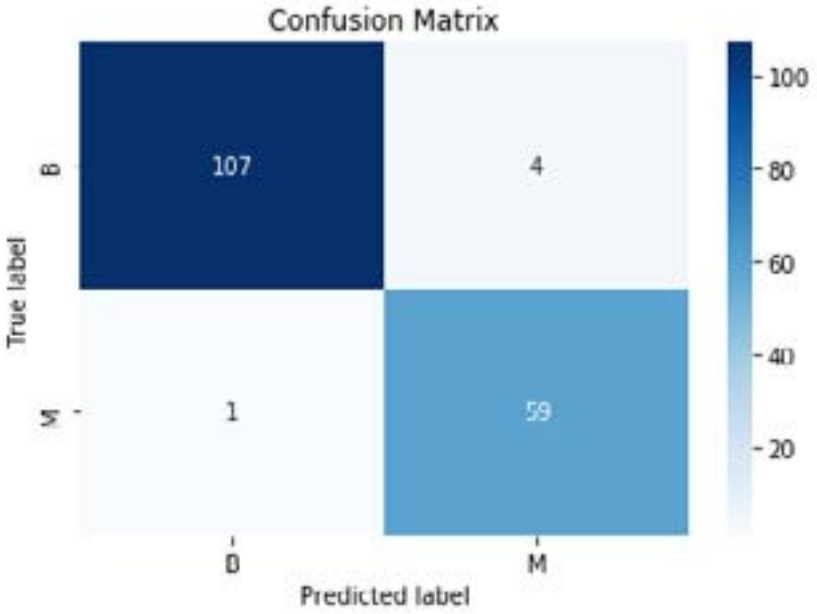
```
prediction_rf=model_rf.predict(X_test)
print('The accuracy of the Random Forest classifier is',metrics.accuracy_score(prediction_rf,y_test))
```

```
The accuracy of the Random Forest classifier is 0.9707602339181286
```

This is by far the best accuracy with 97% and let's see if changes in important features affects the accuracy. With classification report we can find out which class got higher accuracy.

	precision	recall	f1-score	support
B	0.99	0.96	0.98	111
M	0.94	0.98	0.96	60
micro avg	0.97	0.97	0.97	171
macro avg	0.96	0.97	0.97	171
weighted avg	0.97	0.97	0.97	171

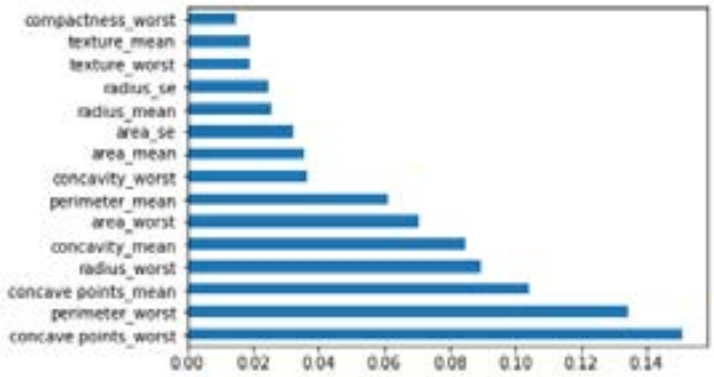
Confusion Matrix of Random Forest Classifier



> **Top 15** important feature according to Random forest

```
feat_importances = pd.Series(model_rf.feature_importances_, index=features.columns)
feat_importances.nlargest(15).plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1169c630>
```



We can see that concave points worst, concave mean and perimeter worst, radius worst holds the most importance in features of this dataset.

With the feature importance , now we can use the top 15 features to model and leave the rest to model on random forest classifier.

```
model=RandomForestClassifier(n_estimators=50,min_samples_split=4)
model.fit(train_X,train_y)
prediction = model.predict(test_X)
metrics.accuracy_score(prediction,test_y)
```

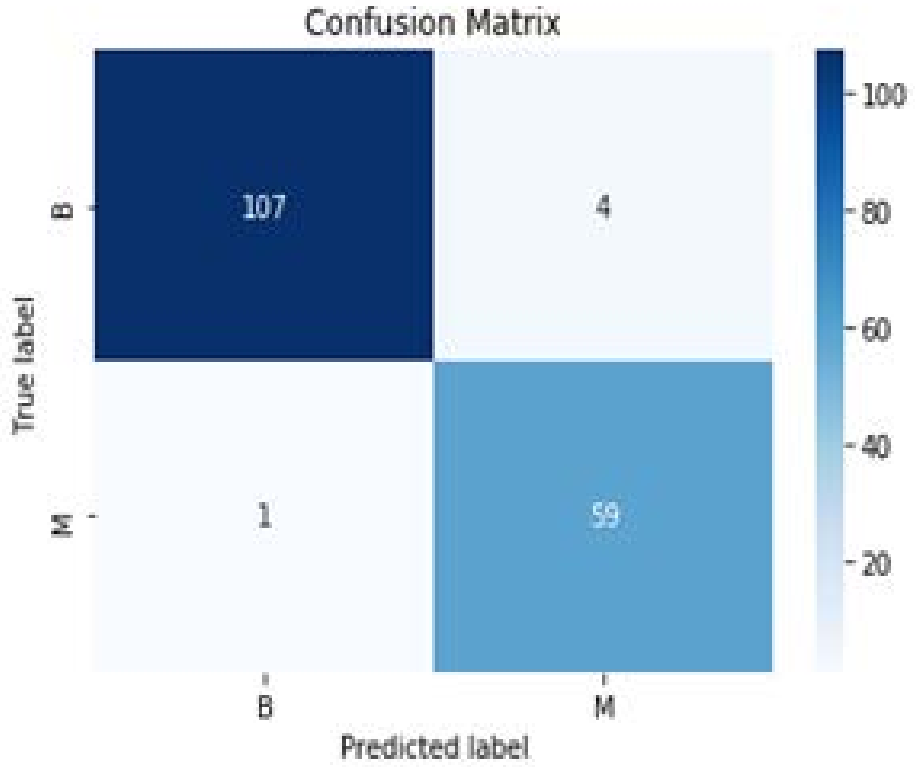
```
0.9766081871345029
```

There is a boost of 0.05% using the top features of random forest lets look at the classification report to find out which class does well in accuracy.

	precision	recall	f1-score	support
B	0.99	0.96	0.98	111
M	0.94	0.98	0.96	60
micro avg	0.97	0.97	0.97	171
macro avg	0.96	0.97	0.97	171
weighted avg	0.97	0.97	0.97	171

CONFUSION MATRIX - with top 15 features on Random forest

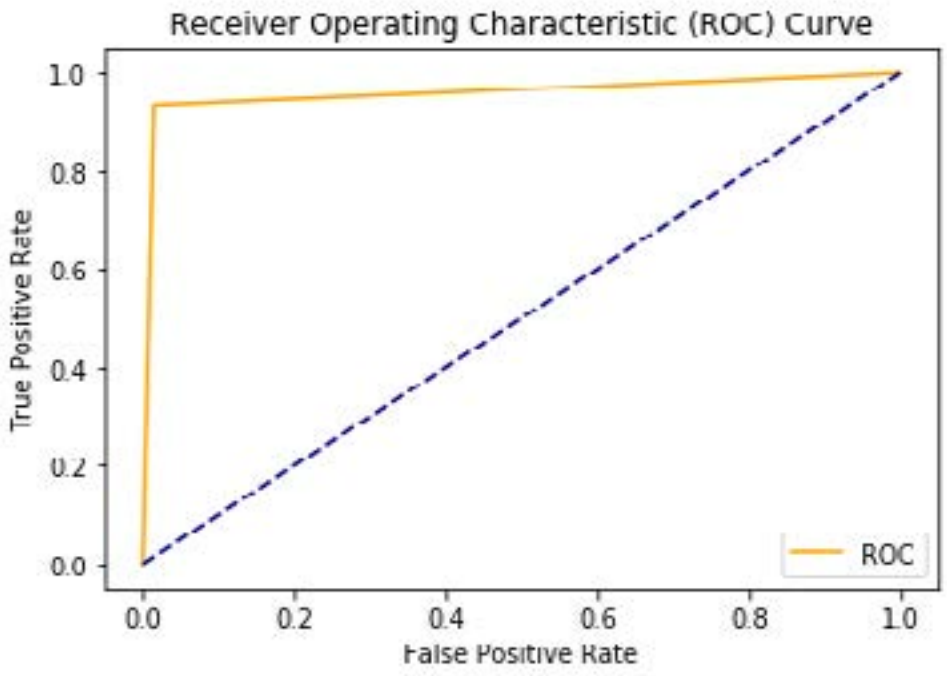
Confusion matrix with top 15 features on Random forest –
Accuracy is same as before and the TP,TN,FP,FN values are same as normal random forest classifier.



AUC – ROC curve

AUC–ROC curve is the model selection metric for bi–multi class classification problem. ROC is a probability curve for different classes. ROC tells us how good the model is for distinguishing the given classes, in terms of the predicted probability. The area covered by the curve is the area between the orange line (ROC) and the axis. This area covered is AUC. The bigger the area covered, the better the machine learning models is at distinguishing the given classes. Ideal value for AUC is 1.

> A typical ROC curve has False Positive Rate **(FPR)** on the X-axis and True Positive Rate **(TPR)** on the Y-axis.



Further Proceedings

- Although this dataset is small and had more features, if there is more data, model would have learned more about the features.
- As this dataset is so crucial and important in the science industry, there should be more contributors for the dataset which leads in better ml model.

ABOUT US

Pepgra is a leading global contract research outsourcing organization provider of scientific, knowledge-based services to bio-pharmaceutical, generic pharmaceutical, biotech, medical device companies and healthcare companies in the areas of clinical trial monitoring, regulatory writing, post-market surveillance, biostatistics and statistical programming services. Our mission is to become a strategic partner to global life science companies providing high quality knowledge-based expertise across the product lifecycle with the ultimate objective of improving quality of healthcare for patients worldwide. Our corporate headquarter is located in India with operations in USA, and the Philippines

Format type: E-Book

UK: 10 Park Place,
Manchester M4 4EY.
UK: +44-1143520021
Email:info@pepgra.com
Web:www.pepgra.com