

In [207]: `%cd C:\Users\manoj\Downloads`

C:\Users\manoj\Downloads

In [208]: `import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import scipy.stats`

In [209]: `df=pd.read_csv('diabetes.csv')`

In [210]: `df.head()`

Out[210]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [211]: `X=np.array(df.iloc[:,8])`

In [212]: `y=np.array(df.iloc[:,8])`

In [213]: `df['Outcome'].value_counts()`

Out[213]: 0 500
1 268
Name: Outcome, dtype: int64

In [214]:

```
from numpy import mean
from numpy import std
from sklearn.datasets import make_classification
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
```

In [215]: `cv = KFold(n_splits=10, random_state=1, shuffle=True)`In [216]: `model = LogisticRegression(max_iter=500)`In [217]: `scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)`In [218]: `print('Accuracy: %.2f (%.2f)' % (mean(scores), std(scores)))`

Accuracy: 0.77 (0.04)

In [219]:

```
from scipy.stats import sem
from numpy import mean
from numpy import std
from sklearn.datasets import make_classification
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from matplotlib import pyplot
```

In [220]: `cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)`In [221]: `model = LogisticRegression(max_iter=500)`In [222]: `scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)`

```
In [223]: print('Accuracy: %.2f (%.2f)' % (mean(scores), std(scores)))
```

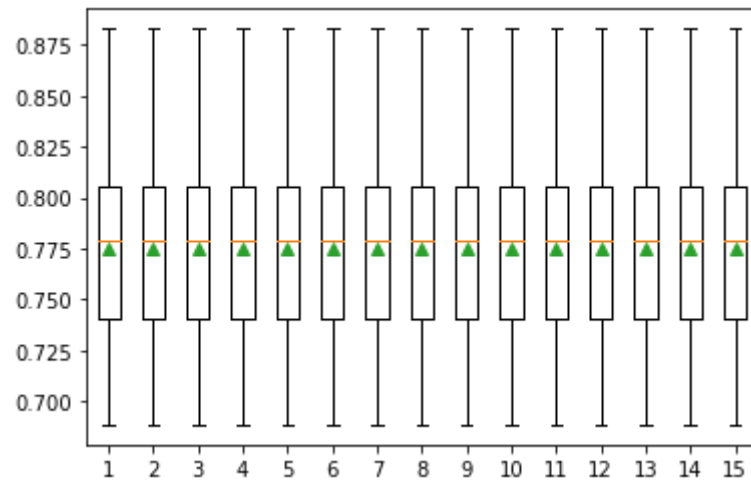
Accuracy: 0.78 (0.05)

```
In [224]: def evaluate_model(X, y, repeats):  
    cv = RepeatedKfold(n_splits=10, n_repeats=5, random_state=1)  
    model = LogisticRegression(max_iter=500)  
    scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)  
    return scores
```

```
In [225]: repeats = range(1,16)  
results = list()  
for r in repeats:  
    scores = evaluate_model(X, y, r)  
    print('>%d mean=%.4f se=%.4f' % (r, mean(scores), sem(scores)))  
    results.append(scores)
```

```
>1 mean=0.7752 se=0.0070  
>2 mean=0.7752 se=0.0070  
>3 mean=0.7752 se=0.0070  
>4 mean=0.7752 se=0.0070  
>5 mean=0.7752 se=0.0070  
>6 mean=0.7752 se=0.0070  
>7 mean=0.7752 se=0.0070  
>8 mean=0.7752 se=0.0070  
>9 mean=0.7752 se=0.0070  
>10 mean=0.7752 se=0.0070  
>11 mean=0.7752 se=0.0070  
>12 mean=0.7752 se=0.0070  
>13 mean=0.7752 se=0.0070  
>14 mean=0.7752 se=0.0070  
>15 mean=0.7752 se=0.0070
```

```
In [226]: pyplot.boxplot(results, labels=[str(r) for r in repeats], showmeans=True)  
pyplot.show()
```



```
In [ ]:
```