Gaussian Naive Bayes Classifier

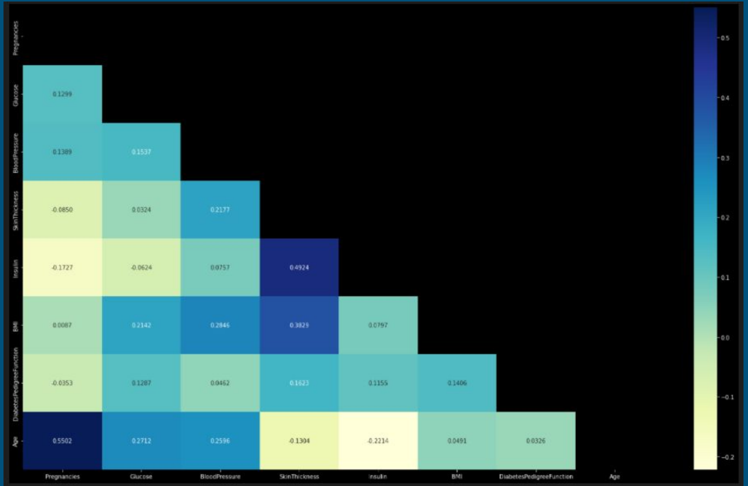
Charlie Rehder, Luis Rivera, Parth Patel, Marjea Mckoy

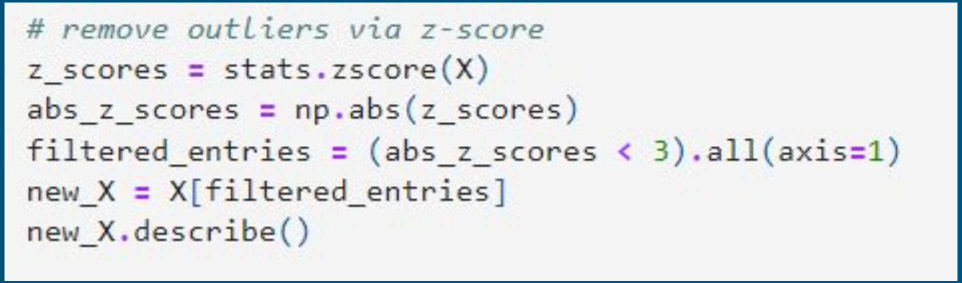
Overview

- Pima Indians Diabetes Dataset - numeric, continuous data
- Target variable is a yes/no value
- Calls for supervised, classifier algorithm
- Gaussian classifier would work best with dataset
- Explain how Gaussian differs from Monomial Classifier

Data Processing

- Data set had no null values
- All data types were converted to floats
- Outliers were removed (value is an outlier if it is 3 standard deviations away from the mean)
- No strong correlation was found between columns, so none were removed

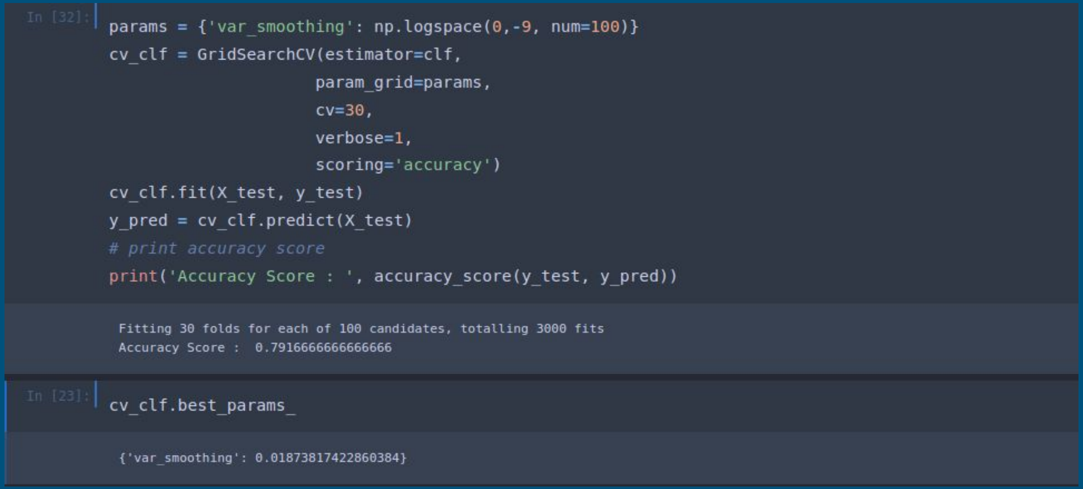




```
# remove outliers via z-score
z_scores = stats.zscore(X)
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3).all(axis=1)
new_X = X[filtered_entries]
new_X.describe()
```

Data Processing (cont...)

- **var_smoothing** was the only hyperparameter that needed tuning
- Done so that no probabilities are zero
- Using a grid search we found the optimal value to be **0.019**
- Accuracy score went slightly up from **0.765** to **0.792**



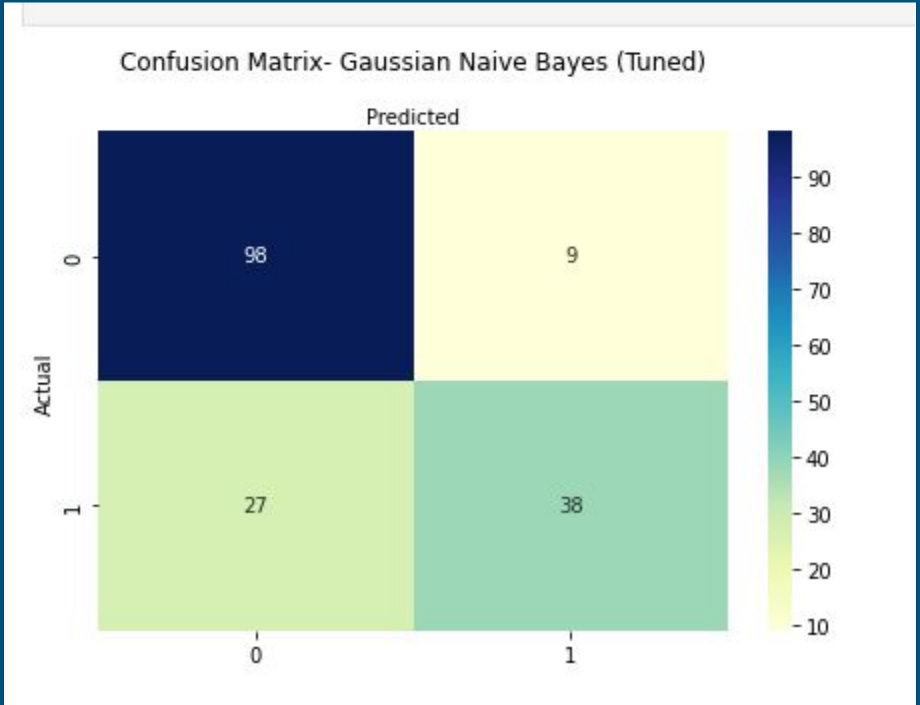
```
In [32]:
params = {'var_smoothing': np.logspace(0,-9, num=100)}
cv_clf = GridSearchCV(estimator=clf,
                      param_grid=params,
                      cv=30,
                      verbose=1,
                      scoring='accuracy')
cv_clf.fit(X_test, y_test)
y_pred = cv_clf.predict(X_test)
# print accuracy score
print('Accuracy Score : ', accuracy_score(y_test, y_pred))
```

```
Fitting 30 folds for each of 100 candidates, totalling 3000 fits
Accuracy Score :  0.7916666666666666
```

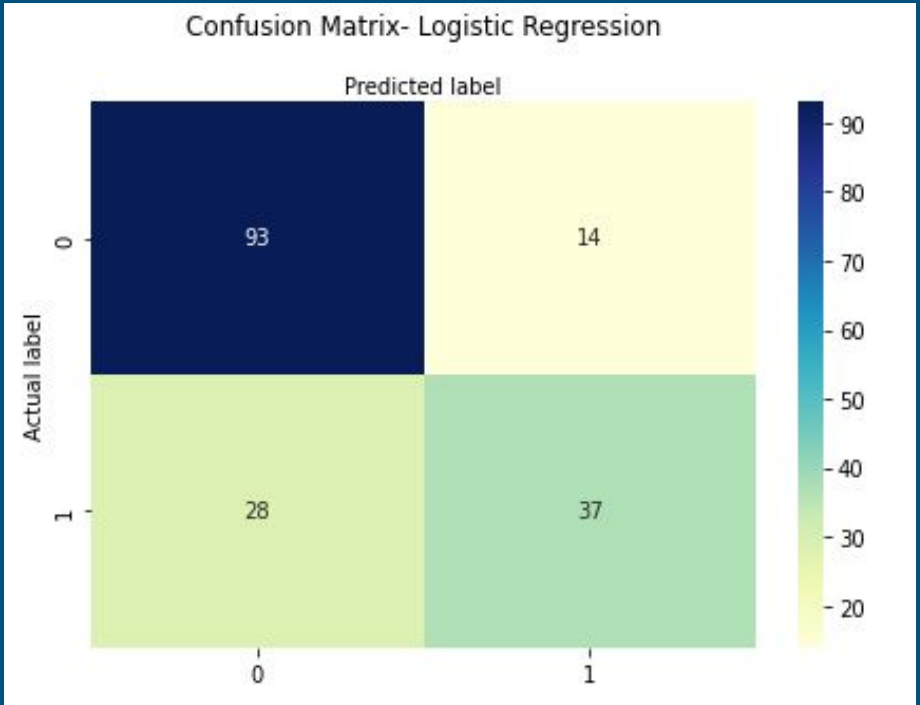
```
In [23]:
cv_clf.best_params_
```

```
{'var_smoothing': 0.018738174228603844}
```

Results


Confusion Matrix- Gaussian Naive Bayes (Tuned)

Model score: 0.79


Confusion Matrix- Logistic Regression

Model score: 0.75

Ways to Improve

- **Convert to log probabilities** - better computational representation of smaller probabilities
- **Train with larger dataset** - further generalize model
- **Handle zero probabilities** - data smoothing
- **Building model in parallel** - doesn’t improve performance but model can be built quickly, each probability is computed independently