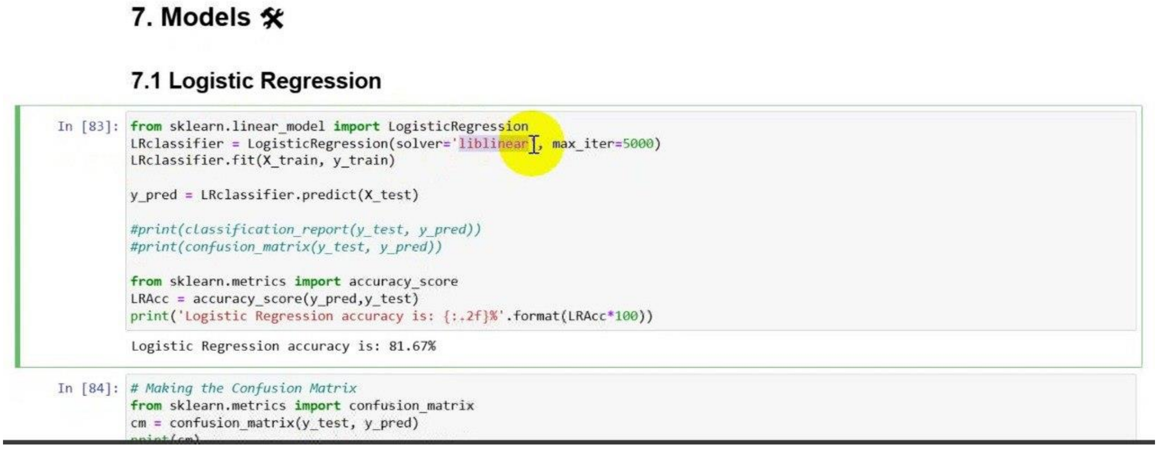

7. Models 🛠

7.1 Logistic Regression

```
In [83]: from sklearn.linear_model import LogisticRegression
         LRclassifier = LogisticRegression(solver='liblinear', max_iter=5000)
         LRclassifier.fit(X_train, y_train)

         y_pred = LRclassifier.predict(X_test)

         #print(classification_report(y_test, y_pred))
         #print(confusion_matrix(y_test, y_pred))

         from sklearn.metrics import accuracy_score
         LRAcc = accuracy_score(y_pred,y_test)
         print('Logistic Regression accuracy is: {:.2f}%'.format(LRAcc*100))

         Logistic Regression accuracy is: 81.67%

In [84]: # Making the Confusion Matrix
         from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test, y_pred)
         print(cm)
```

Solver linear is a feature in spreadsheet software like Microsoft Excel or Google Sheets that allows you to solve linear optimization problems. It's a tool for finding the optimal solution to a problem with linear constraints and objective function.

The Solver linear function can be used to:

1. Maximize or minimize a linear objective function
2. Subject to linear constraints
3. Find the values of the decision variables that satisfy the constraints and optimize the objective function

For example, you could use Solver linear to find the optimal product mix that maximizes profit while respecting production capacity constraints. Or you could use it to determine the optimal investment portfolio that maximizes return while limiting risk.

The Solver linear function uses mathematical optimization algorithms, such as the Simplex method or the Gradient Descent method, to find the optimal solution. It allows you to quickly explore different scenarios and find the best decision given your objectives and constraints.

Overall, Solver linear is a powerful tool for solving complex linear programming problems in a spreadsheet environment. It can save time and improve decision-making across a wide range of business and optimization applications.