Logistic Regression

Prajjwal Vijaywargiya - 2017B3A70954H
Siddhi Mahesh Burse - 2017B3A70972H
Parth Krishna Sharma - 2017B3A70907H

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
```

In [2]:
```
train = pd.read_csv("D:/BITS/DL/Assignments/Assignment1/mnist_train.csv")
test = pd.read_csv("D:/BITS/DL/Assignments/Assignment1/mnist_test.csv")
```

In [3]:
```
X_train = train.iloc[:,1:].values
y_train = train.iloc[:,0].values
```

In [4]:
```
X_test = test.iloc[:,1:].values
y_test = test.iloc[:,0].values
```

In [5]:
```
X_train,X_test = X_train/255.0,X_test/255.0
```

In [6]:
```
lr = LogisticRegression(max_iter=1000,verbose=2)
lr.fit(X_train,y_train)
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:  3.4min remaining:    0.0s
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:  3.4min finished
```

Out[6]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=2,
                   warm_start=False)
```

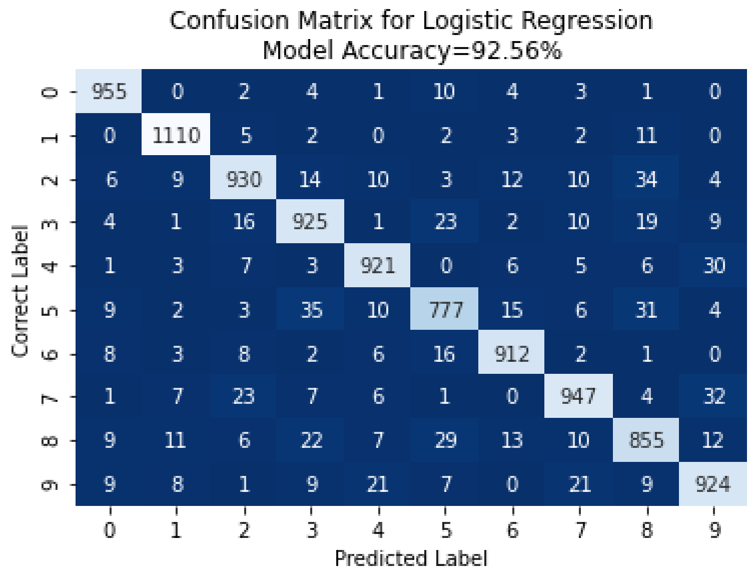
In [7]:
```
pred = lr.predict(X_test)
```

In [8]:
```
acc = accuracy_score(y_test,pred)
acc
```

Out[8]:
```
0.9256
```

In [9]:
```
sns.heatmap(confusion_matrix(y_test,pred),cmap='Blues_r',fmt='',annot=True,cbar=False)
plt.ylabel("Correct Label")
plt.xlabel("Predicted Label")
plt.title("Confusion Matrix for Logistic Regression\nModel Accuracy={}%".format(acc*100))
```

Out[9]:
```
Text(0.5, 1.0, 'Confusion Matrix for Logistic Regression\nModel Accuracy=92.56%')
```



	0	1	2	3	4	5	6	7	8	9
0	955	0	2	4	1	10	4	3	1	0
1	0	1110	5	2	0	2	3	2	11	0
2	6	9	930	14	10	3	12	10	34	4
3	4	1	16	925	1	23	2	10	19	9
4	1	3	7	3	921	0	6	5	6	30
5	9	2	3	35	10	777	15	6	31	4
6	8	3	8	2	6	16	912	2	1	0
7	1	7	23	7	6	1	0	947	4	32
8	9	11	6	22	7	29	13	10	855	12
9	9	8	1	9	21	7	0	21	9	924

In [11]:
```
print("Classification Report for Logistic Regression:\n{}".format(classification_report(y_test,pred)))
```

```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       980
           1       0.96      0.98      0.97      1135
           2       0.93      0.90      0.91      1032
           3       0.90      0.92      0.91      1010
           4       0.94      0.94      0.94       982
           5       0.90      0.87      0.88       892
           6       0.94      0.95      0.95       958
           7       0.93      0.92      0.93      1028
           8       0.88      0.88      0.88       974
           9       0.91      0.92      0.91      1009

    accuracy                           0.93     10000
   macro avg       0.92      0.92      0.92     10000
weighted avg       0.93      0.93      0.93     10000
```

In [ ]: