ML Lab

Manas Agarwal
IBM18CS0F2

```
Class NaiveBayesClassifier:
    def __init__(self, X, Y):
        // initializing
        dim = len(X[0])
        N = len(X)
        attrs = [[] for _ in range(dim)]
        output_dom = {}
        data = []
        for i in range(N):
            for j in range(dim):
                if X[i][j] is not in attrs[j]:
                    attrs[j].append(X[i][j])

            if y[i] in output_dom.keys():
                output_dom[y[i]] = 1

            else
                output_dom[y[i]] += 1
            data.append([X[i], y[i]])
```

Manan Agarwal
IBM18CS015

```
def classify(entry):
    solve = None
    max_arg = -1
    for y in output_dom.keys():
        prob = output_dom[y]/N
        for i in range(dim):
            cases = [x for x in data
                if x[0][i] == entry[i]
                and x[i] == y]
            n = len(cases)
            prob *= n/N
        if prob > max_arg:
            max_arg = prob
            solve = y
    return solve.

nbc = NaiveBayesClassifier(X_train, Y_train)
for i in range(total_cases):
    predict = nbc.classify(X_val[i])
    if y_val[i] == predict:
        corona ✓
    else
        corona X
```