A simple ML alg to predict Cats will be allowed in your house

Import the data as a dataset

In[5]:=
```
maindata = Import[
  "C:\\Users\\gurha\\OneDrive\\Documents\\bhoris\\College\\SYBSC\\stat attack\\dataset.
    csv", "Dataset", "HeaderLines" → 1]
```

Out[5]=

region	price	type	sqfeet	beds
milwaukee	840	apartment	672	1
kalamazoo	935	apartment	720	1
huntsville / decatur	999	apartment	1470	3
flagstaff / sedona	1199	apartment	676	1
wichita	1074	townhouse	1644	4
daytona beach	1379	apartment	1154	2
des moines	655	apartment	682	1
wichita	518	apartment	580	1
new hampshire	1550	apartment	691	1
pensacola	1142	apartment	815	1
chattanooga	625	apartment	620	1
florence	1950	townhouse	1100	2
rochester	1236	townhouse	810	2
ames	1500	apartment	1650	3
hawaii	2595	condo	900	1
billings	572	house	1823	3
birmingham	799	apartment	702	1
topeka	875	apartment	1488	3
southern maryland	1790	apartment	562	0
beaumont / port arthur	730	apartment	750	1

rows 1–20 of 4000 columns 1–10 of 13

Display the column headers

In[6]:=
```
maindata[Union, Keys]
```

Out[6]=

region	price	type	sqfeet	beds	baths	cats_allowed	dogs_allowed	smoking_allowed	co

Define the Objective

In[7]:=
```
objcat = "cats_allowed";
```

Creates a new dataset and assigns an ID to each row

In[8]:=
```
newmaindata = maindata[AssociationThread[Range@Length@#, #] &]
```

Out[8]=

	region	price	type	sqfeet
1	milwaukee	840	apartment	672
2	kalamazoo	935	apartment	720
3	huntsville / decatur	999	apartment	1470
4	flagstaff / sedona	1199	apartment	676
5	wichita	1074	townhouse	1644
6	daytona beach	1379	apartment	1154
7	des moines	655	apartment	682
8	wichita	518	apartment	580
9	new hampshire	1550	apartment	691
10	pensacola	1142	apartment	815
11	chattanooga	625	apartment	620
12	florence	1950	townhouse	1100
13	rochester	1236	townhouse	810
14	ames	1500	apartment	1650
15	hawaii	2595	condo	900
16	billings	572	house	1823
17	birmingham	799	apartment	702
18	topeka	875	apartment	1488
19	southern maryland	1790	apartment	562
20	beaumont / port arthur	730	apartment	750

rows 1–20 of 4000 columns 1–10 of 13

Separates Features and Objective

In[9]:=
```
catdataSplit =
 newmaindata[All, <|"Features" → Values@*KeyDrop[{objcat}], "Objective" -> Key[objcat]|>]
```

Out[9]=

					Featu
1	milwaukee	840	apartment	672	
2	kalamazoo	935	apartment	720	
3	huntsville / decatur	999	apartment	1470	
4	flagstaff / sedona	1199	apartment	676	
5	wichita	1074	townhouse	1644	
6	daytona beach	1379	apartment	1154	
7	des moines	655	apartment	682	
8	wichita	518	apartment	580	
9	new hampshire	1550	apartment	691	
10	pensacola	1142	apartment	815	
11	chattanooga	625	apartment	620	
12	florence	1950	townhouse	1100	
13	rochester	1236	townhouse	810	
14	ames	1500	apartment	1650	
15	hawaii	2595	condo	900	
16	billings	572	house	1823	
17	birmingham	799	apartment	702	
18	topeka	875	apartment	1488	
19	southern maryland	1790	apartment	562	
20	beaumont / port arthur	730	apartment	750	

rows 1–20 of 4000

Make a random training set of 3200 entries and use the remaining as test

In[112]:=
```
numTest = 800;
ids = Range@newmaindata[Length];
testIds = ids ~ RandomSample ~ numTest;
trainIds = ids ~ Complement ~ testIds;
dataUnclass = catdataSplit[<|"Test" → testIds, "Train" → trainIds|>];
```

Training the classifier

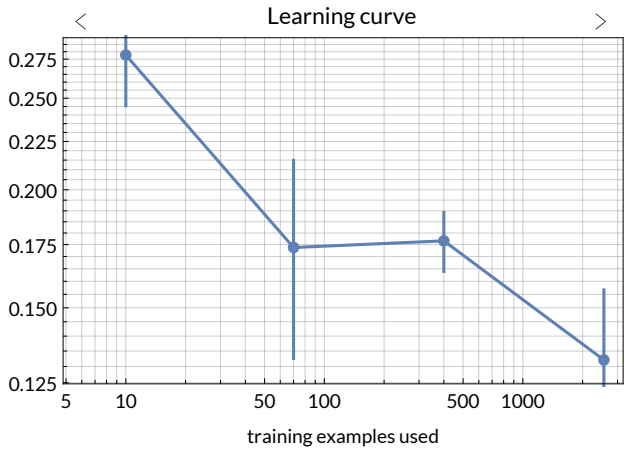
In[117]:=
```
classifiertrain = dataUnclass["Train", Classify@*Values, #Features → #Objective &];
```

In[118]:=
```
Information[classifiertrain]
```

Out[118]=

Classifier information

Data type	Mixed (number: 12)
Classes	0, 1
Accuracy	(96.1 ± 1.1)%
Method	LogisticRegression
Single evaluation time	9.7 ms/example
Batch evaluation speed	5.43 examples/ms
Loss	0.132 ± 0.025
Model memory	418. kB
Training examples used	3200 examples
Training time	9.09 s



Appending classifications in test dataset

In[119]:=
```
dataClass = dataUnclass[
    {"Test" → Query[All, Append[#, <|"Classified as" → classifiertrain@#Features|>] &]}];
```

Performance

In[120]:=
```
cfm = dataUnclass["Test",
    ClassifierMeasurements[classifiertrain, #] &@*Values, #Features → #Objective &];
cfm["Accuracy", ComputeUncertainty → True]
```

Out[121]= 0.945 ± 0.008

In[122]:=
```
cfm["Precision"]
```

Out[122]= ⟨| 0 → 0.864407, 1 → 0.978723 |⟩

In[123]:=
```
cfm["Specificity"]
```

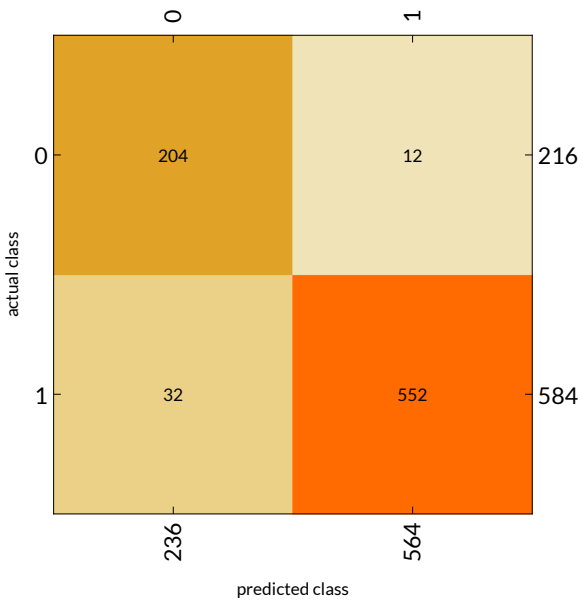
Out[123]= ⟨| 0 → 0.945205, 1 → 0.944444 |⟩

In[124]:=
```
cfm["Error"]
```

Out[124]= 0.055

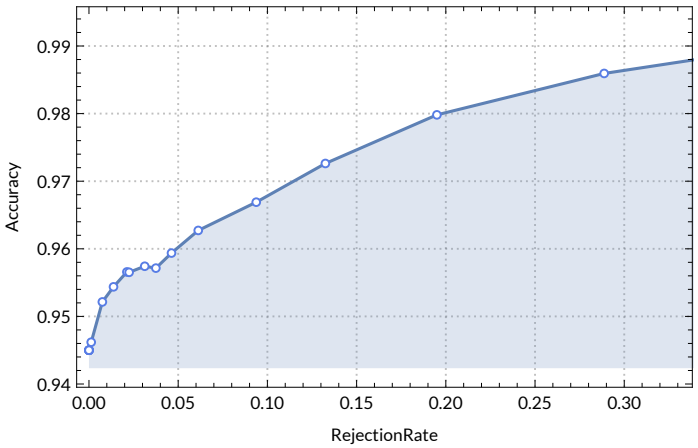
In[125]:=
```
cfm["ConfusionMatrixPlot"]
```

Out[125]=



In[126]:=
```
cfm["AccuracyRejectionPlot"]
```

Out[126]=



In[127]:=
```
cfm["ROCCurve"]
```

Out[127]=