SEARCH

RESOURCES

CONCEPTS

Knowledge

Get learning questions answered

Student Hub

Chat with peers and mentors

2. Implement the `predict(observation)` method in `classifier.cpp`.

   Given a new data point, prediction requires two steps:

   1. **Compute the conditional probabilities for each feature/label combinat** and label C with mean μ and standard deviation σ (computed in traini probability can be computed using the formula here:

   $$p(x=v|C) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(v-\mu)^2}{2\sigma^2}}$$

   Here v is the value of feature x in the new data point.

   2. **Use the conditional probabilities in a Naive Bayes classifier.** This can formula here:

   $$y = \underset{k\in(1,\ldots,K)}{argmax}\ p(C_k)\prod_{i=1}^{n} p(x_i = v_i|C_k)$$

   In this formula, the argmax is taken over all possible labels C_k and the all features x_i with values v_i.

3. When you want to test your classifier, run `Test Run` and check out the res

NOTE: You are welcome to use some existing implementation of a Gaussian Naive to get the **best** results you will still need to put some thought into what **features** y algorithm when classifying. Though you will only be given the 4 coordinates listed that by "engineering" features you may get better performance. For example: the coordinate may not be that useful. But `d % lane_width` might be helpful since position of a vehicle in it's lane regardless of which lane the vehicle is in.

Helpful Resources

- sklearn documentation on GaussianNB
- Wikipedia article on Naive Bayes / GNB

Extra Practice

Provided in one of the links below is `python_extra_practice`, which is the sar written in Python that you can optionally go through for extra coding practice. The available at the `python_solution` link. If you get stuck on the quiz see if you ca solution to C++ and pass the classroom quiz with it. The last link `Nd013_Pred_Da` training and testing data for this problem in case you want to run the problem off

main.cpp classifier.cpp classifier.h

```
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
#include "classifier.h"

using std::cout;
using std::endl;
using std::ifstream;
using std::string;
using std::vector;

// Helper functions to load .txt files
vector<vector<double> > Load_State(string file_name);
```