**BILKENT UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

# CS464

# Introduction to Machine Learning - Fall 2021

## Homework 1

Bulut Gözübüyük 21702771

# 1 Probability

$P(heads|A) = P1$

$P(heads|B) = P2$

$1 - P(A) = P(B)\ \ P(A) = P3$

$P(heads|A) = \frac{P(heads, A)}{P(A)} = P1 = \frac{P(heads,A)}{P3}$

$P(heads) = P(A) * P(heads|A) + P(B) * P(heads|B)$

## Question 1.1

$(1 - P(heads, A))^7 * P(heads, A) = (1 - P1 * P3)^7 * P1 * P3$

## Question 1.2

$P(heads) = P3 * P1 + (1 - P3) * P2 = P3 * P1 + P2 - P2P3$

$E(x) = \sum_{i=1}^{10} x_i p(x_i) = 10 * (P3 * P1 + P2 - P2P3)$

## Question 1.3

### Question 1.3a

Trials is the key of measuring the performance of Oliver. If trial count increases, data becomes closer to real guessing performance of Oliver. We can examine the validity of the probabilities by comparing it with the probabilities calculated from given sample ones.

## Question 1.3b

8 heads in a row probability:

P(“he tells you that you’ll not obtain heads next time, you will not obtain heads.”) = 0.99

P(heads|”he tells you that you’ll not obtain heads next time, you will not obtain heads.”) = 0.95

$$P("Oliver\ predicting\ heads\ in\ a\ row")^{8} = (0.99)^{8} = 0.923$$

$$P("Oliver\ predicting\ true\ heads\ in\ a\ row")^{8} = (0.99\ *\ 0.95)^{8} = 0.612$$

## Question 1.3c

T = “he tells you that you’ll not obtain heads next time, you will not obtain heads.”

$$P(heads)\ =\ P(heads|T)P(T)\ +\ P(heads|"guessed\ not\ heads")P("guessed\ not\ heads")$$
$$=\ (0.99\ *\ 0.95)\ +\ 0.01\ *\ 0.01\ =\ 0.9406$$

$$P("Oliver\ guesses\ not\ heads"|heads)\ = \frac{P(heads|"Oliver\ guesses\ not\ heads")P("Oliver\ guesses\ not\ heads")}{P(heads)}$$

$$= \frac{0.01*0.01}{0.9406} = 1.06\ *\ 10^{-4}$$

# 2 kNN Diabetes Classifier

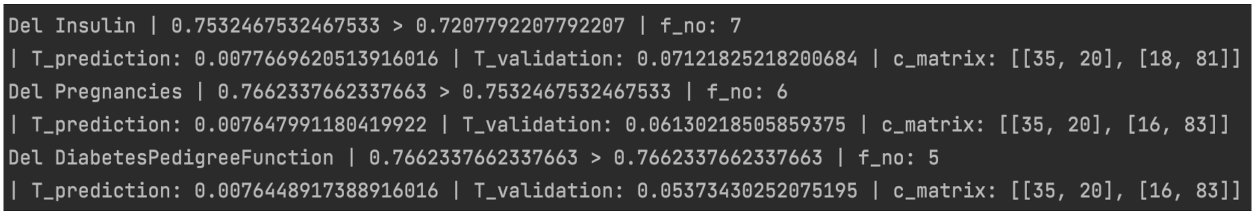
### Question 2.1

As a distance metric, we should select Euclidian distance since it calculates the distance between two places in space on a regular basis.

### Question 2.2

It is not always the case that using more features will result in better accuracy. We should select the best ‘m’ features from ‘W’ in order to make the accuracy better.

### Question 2.3



```
Del Insulin | 0.7532467532467533 > 0.7207792207792207 | f_no: 7
| T_prediction: 0.0077669620513916016 | T_validation: 0.07121825218200684 | c_matrix: [[35, 20], [18, 81]]
Del Pregnancies | 0.7662337662337663 > 0.7532467532467533 | f_no: 6
| T_prediction: 0.007647991180419922 | T_validation: 0.06130218505859375 | c_matrix: [[35, 20], [16, 83]]
Del DiabetesPedigreeFunction | 0.7662337662337663 > 0.7662337662337663 | f_no: 5
| T_prediction: 0.0076448917388916016 | T_validation: 0.05373430252075195 | c_matrix: [[35, 20], [16, 83]]
```

| | Accuracy after deletion | Training Time | Validation time |
|---|---|---|---|
| Delete Insulin | 0.7532467532467533 | 0.0077669620513916016 | 0.07121825218200684 |
| Delete Pregnancies | 0.7662337662337663 | 0.007647991180419922 | 0.06130218505859375 |
| Delete DiabetesPedigreeFunction | 0.7662337662337663 | 0.0076448917388916016 | 0.05373430252075195 |

## Question 2.4

From the previous table in Q2.3, it can be observed that training time stays almost similar after deletion steps. Theoretically, training time should be less since there is less features in dataset. This may be caused by the operating system of computer.

Validation time decreases after each step of deletion. This is normal too since there are less features to process.

# 3 Spam SMS Message Detection

### Question 3.1

Multinomial accuracy: 0.9550102249488752

Confusion matrix:

```
[[124, 16],
 [28, 810]]
```

### Question 3.2

3458 Features
2 Labels

P(ham) = 1 - P(spam)

Number of parameters we need to estimate for this model = Feature Count * Label count + 1

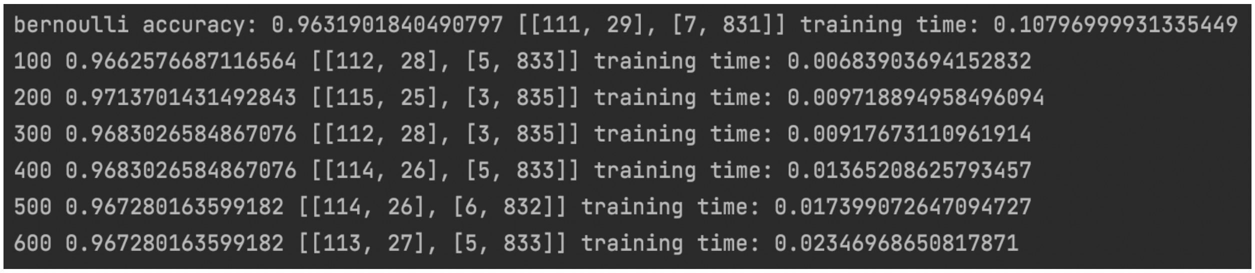
= 3458 * 2 + 1 = 6917

## Question 3.3

Mutual Information Formula[1]

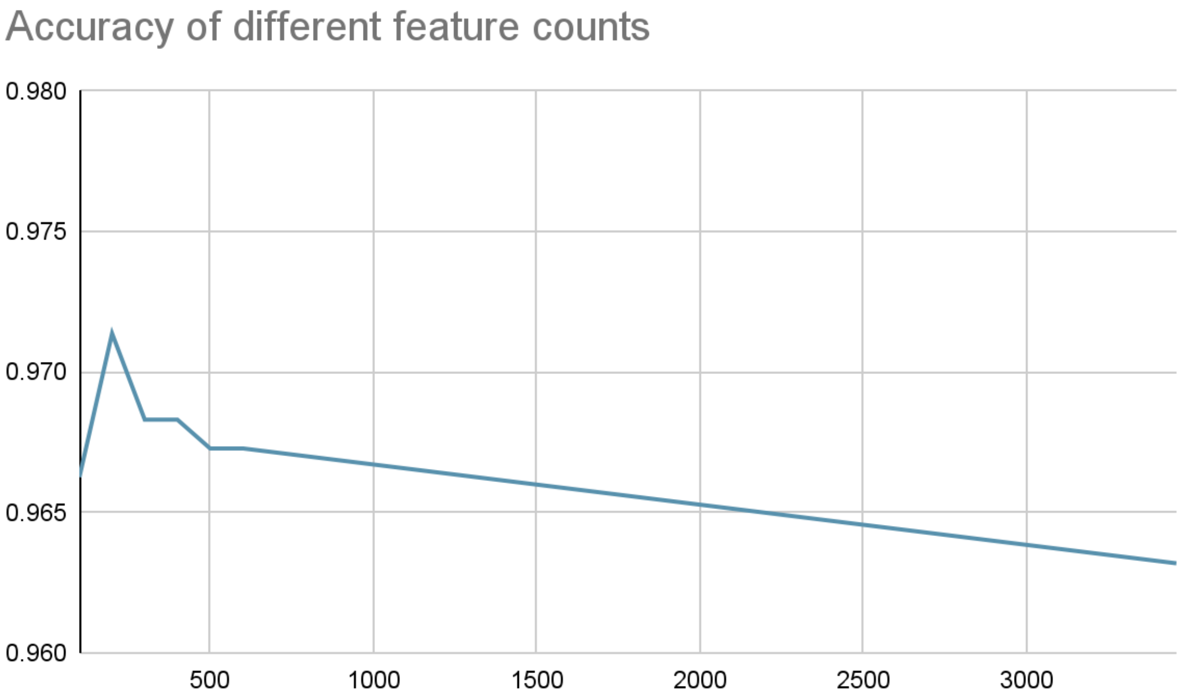
$$(13.17) \qquad I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

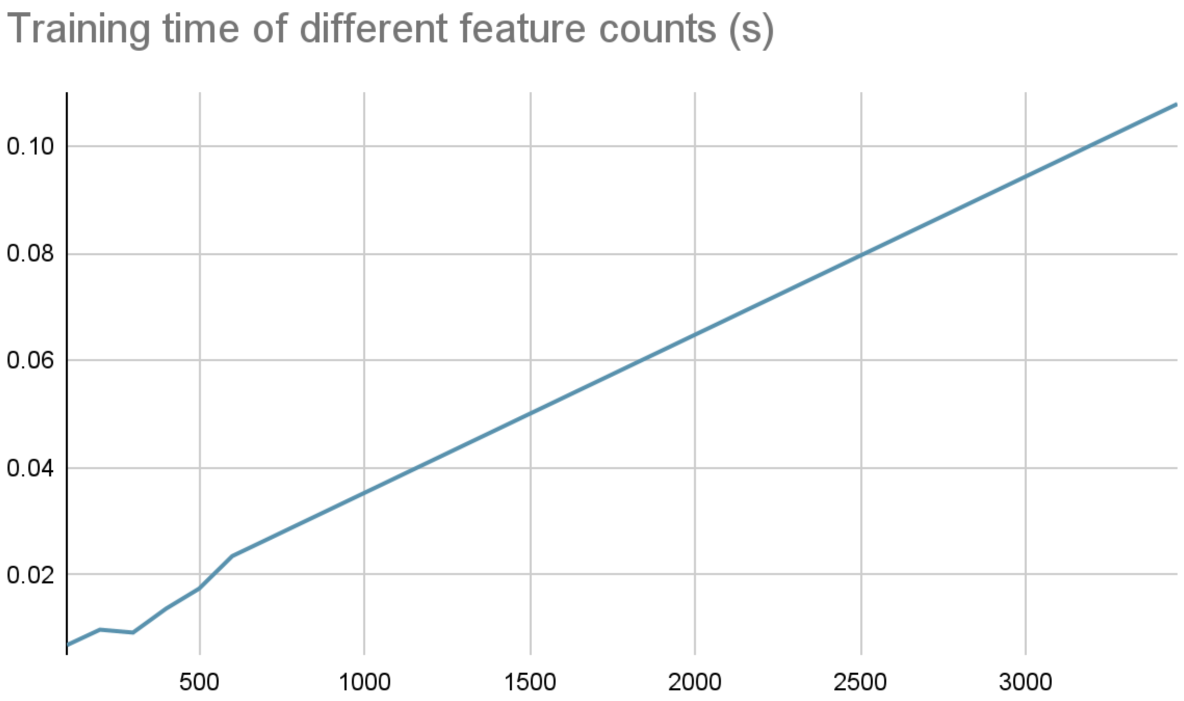
### Question 3.3a



```
bernoulli accuracy: 0.9631901840490797 [[111, 29], [7, 831]] training time: 0.10796999931335449
100 0.9662576687116564 [[112, 28], [5, 833]] training time: 0.00683903694152832
200 0.9713701431492843 [[115, 25], [3, 835]] training time: 0.009718894958496094
300 0.9683026584867076 [[112, 28], [3, 835]] training time: 0.00917673110961914
400 0.9683026584867076 [[114, 26], [5, 833]] training time: 0.01365208625793457
500 0.967280163599182 [[114, 26], [6, 832]] training time: 0.017399072647094727
600 0.967280163599182 [[113, 27], [5, 833]] training time: 0.02346968650817871
```

Best accuracy: F = 200    0.971370143192843
Confusion matrix is available after accuracy in the output picture.
Accuracy seems to decreases with feature count.





It can be seen that, training time increases with increasing feature counts.

## Question 3.3b

Since training time increases after each step, it can be concluded that it has effect on time complexity. If feature count increases then time complexity increases.

## Question 3.4

| | Accuracy |
|---|---|
| Multinomial Naive Bayes | 0.955 |
| Bernoulli Naive Bayes | 0.963 |
| Bernoulli F = 200 | 0.971 |

According to table, it can be observed that even though default feature count, bernoulli performs better in terms of accuracy. After selecting 200 best features, bernoulli becomes much accurate.

# References

[1] “Introduction to information retrieval - stanford university.” [Online]. Available: https://nlp.stanford.edu/IR-book/pdf/13bayes.pdf. [Accessed: 07-Nov-2021].