**Advanced Data Science Homework 2**

**1.Probability**

1.

(a) 

(b)



(c)



2.

(a)

We notice that  plugging in the correlation definition we get  , we'll prove this using Cauchy-Schwarz.



(b)

Using the CS property that equality holds iff  we get that  iff  ,since  we'll examine cov(X,Y):



And since Var(X)>X we're getting that for



(c)

Yes, proof: From Q1 section (a) (\*) we can conclude that if I(X,Y)=0 => P(X,Y)=P(X)P(Y) i.e. X and Y are statistically independent thus:



Which concludes the proof

(d)

No, counter example: let  and let  ,



On the other hand:



Thus X and Y are not necessarily statistically independent

**2.** **Generative and Discriminative Classifiers: Gaussian (Naive) Bayes and Logistic Regression**

**2.1**

The answer is no, the new form is not the same one used in logistic regression, proof:

In the lecture we saw that  =(\*)

Introducing the new – no assumption – model we'll get



Plugging that to the form (\*) we get

 which is obviously not in the form of logistic regression (no linear separating surface).

**2.2**

The answer is yes. To see this, we need to expand the formula for (there is no need to look at Y=0 under the assumption of binary classification).

Using Bayes rule and expanding:

Let us focus only on the probabilities for now

Again, we are not interested in the probability of Y=0/1 as we know it is a Bernoulli distribution and as such a constant.

Finally, we found that we get an expression with some weights on the X. and if we would insert back this result into the previous equations we would find

Which is like logistic regression.

**3.2 Data Cleaning**

For the first part of the question – visualization of the data and comparing train to test and Hilary to Trump, we didn't need data cleaning as we used features we thought relevant and those were ok. We did used smart feature which we formed manually to test their distributions, and by that procedure decided which features to include in the classifier.

In the classification part, we encountered problems as some features were NaN at some of the tweets. To solve this we replaced the NaN values for a proper value i.e. in the original\_author feature there isn't much sense in "smart" filling of those values, thus we used an empty string.