Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

Task 1

Vocabulary = { happy, new, year, holiday, term, starts, work, celebrations }
|Vocabulary| = 8

Prior:

$P(c_1) = \frac{3}{5} = 0.6$
$P(c_2) = \frac{2}{5} = 0.4$

Posterior: (with Add-One-Smoothing)

$P(happy|c_1) = \frac{2+1}{7+8} = \frac{3}{15}$
$P(new|c_1) = \frac{2+1}{7+8} = \frac{3}{15}$
$P(year|c_1) = \frac{2+1}{7+8} = \frac{3}{15}$
$P(holiday|c_1) = \frac{1+1}{7+8} = \frac{2}{15}$
$P(term|c_1) = \frac{0+1}{7+8} = \frac{1}{15}$
$P(starts|c_1) = \frac{0+1}{7+8} = \frac{1}{15}$
$P(work|c_1) = \frac{0+1}{7+8} = \frac{1}{15}$
$P(celebrations|c_1) = \frac{0+1}{7+8} = \frac{1}{15}$

$P(happy|c_2) = \frac{0+1}{4+8} = \frac{1}{12}$
$P(new|c_2) = \frac{0+1}{4+8} = \frac{1}{12}$
$P(year|c_2) = \frac{0+1}{4+8} = \frac{1}{12}$
$P(holiday|c_2) = \frac{0+1}{4+8} = \frac{1}{12}$
$P(term|c_2) = \frac{1+1}{4+8} = \frac{2}{12}$
$P(starts|c_2) = \frac{2+1}{+}14 + 8 = \frac{3}{12}$
$P(work|c_2) = \frac{1+1}{4+8} = \frac{2}{12}$
$P(celebrations|c_2) = \frac{0+1}{4+8} = \frac{1}{12}$

$P(c_1|d) = \frac{3}{5} * \frac{3}{15} * \frac{3}{15} * \frac{3}{15} * \frac{1}{15} = \frac{81}{253125} = c_{map}$
$P(c_2|d) = \frac{1}{12} * \frac{1}{12} * \frac{1}{12} * \frac{1}{12} = \frac{2}{103680}$

The model assigned the class c_1 to the document.

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

Task 2

Task 2: Maximum Entropy Classification

2.1)

$$P_\lambda(SPAM|x_1) = \frac{\exp\left(\sum_i \lambda_i \cdot f_i(SPAM, x_1)\right)}{\sum_{y'} \exp\left(\sum_i \lambda_i \cdot f_i(y', x_1)\right)}$$

$$= \frac{\exp(0{,}2 \cdot 1 + 0 + 0{,}5 \cdot 1 + 0 + 0 + 0 + 0{,}1 \cdot 1 + 0)}{\exp(0{,}2+0{,}5+0{,}1) + \exp(-0{,}1 \cdot 1 + (-0{,}2) \cdot 1 + 0 \cdot 1)}$$

$$= \frac{\exp(0{,}8)}{\exp(0{,}8) + \exp(-0{,}3)}$$

$$\approx \frac{2{,}22554}{2{,}22554 + 0{,}74081}$$

$$= \frac{2{,}22554}{2{,}96635} \approx 0{,}75026$$

2,2)

$$\frac{A_\lambda}{\partial \lambda_6} = \sum_{(y,x) \in (Y,X)} f_6(y, x)$$

$$= 0+0+0+0+1+1 = 2$$

$$\frac{B_\lambda}{\partial \lambda_6} = \sum_{(y,x) \in (Y,X)} \sum_{y'} P_\lambda(y'|x) \cdot f_6(y', x)$$

$$= P_\lambda(SPAM|x_1) \cdot f_6(SPAM, x_1) + P_\lambda(HAM|x_1) \cdot f_6(HAM, x_1)$$
$$+ P_\lambda(SPAM|x_2) \cdot f_6(SPAM, x_2) + P_\lambda(HAM|x_2) \cdot f_6(HAM, x_2)$$
$$+ P_\lambda(SPAM|x_3) \cdot f_6(SPAM, x_3) + P_\lambda(HAM|x_3) \cdot f_6(HAM, x_3)$$
$$+ P_\lambda(SPAM|x_4) \cdot f_6(SPAM, x_4) + P_\lambda(HAM|x_4) \cdot f_6(HAM, x_4)$$
$$+ P_\lambda(SPAM|x_5) \cdot f_6(SPAM, x_5) + P_\lambda(HAM|x_5) \cdot f_6(HAM, x_5)$$
$$+ P_\lambda(SPAM|x_6) \cdot f_6(SPAM, x_6) + P_\lambda(HAM|x_6) \cdot f_6(HAM, x_6)$$

$$= P_\lambda(HAM|x_5) + P_\lambda(HAM|x_6)$$

$$= \frac{\exp(-0{,}2+0{,}4+0)}{\exp(0{,}2) + \exp(0{,}5 + (-0{,}1) + 0{,}1)} + \frac{\exp(-0{,}1+0{,}4+0)}{\exp(0{,}3) + \exp(0{,}2 + (-0{,}1) + 0)}$$

$$= \frac{\exp(0{,}2)}{\exp(0{,}2) + \exp(0{,}5)} + \frac{\exp(0{,}3)}{\exp(0{,}3) + \exp(0{,}1)}$$

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

$$\approx \frac{1,22140}{1,22140+1,64872} + \frac{1,34986}{1,34986+1,10517}$$

$$= \frac{1,22140}{1,87012} + \frac{1,34986}{1,45503}$$

$$\approx 0,65311 + 0,92772$$

$$= 1,58083$$

$$\frac{\partial A_\lambda}{\partial \lambda_6} - \frac{\partial B_\lambda}{\partial \lambda_6} = 2 - 1,58083 = 0,41917$$

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

Task 3

- k=1
  $P(black|x) = 0$
  $P(white|x) = \frac{1}{1} = 1$
- k=3
  $P(black|x) = \frac{1}{3}$
  $P(white|x) = \frac{2}{3}$
- k=5
  $P(black|x) = \frac{2}{5}$
  $P(white|x) = \frac{3}{5} = 1$

$\frac{3}{5} < \frac{2}{3} < 1$
In case of k=5 the kNN classifier hast the lowest classification confidence.

Programming Task

```python
import nltk
import pandas as pd
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from collections import Counter
from operator import itemgetter

class SVM_Classifier:
    path = "04-assignment/code/"

    def __init__(self):
        self.corpus = self.generate_corpus("games-train.csv")
        self.term_weights_dictionary = self.calc_term_weights_dictionary()



    def generate_corpus(self, filename: str) ->pd.DataFrame:
        """ Take data as an imput and generate the corpus for classification.
            The data-structure is a pandas DataFrame with two columns.
            The labels are converted in binary values(1 = 'gut', -1='schlecht')
            and the text is normalized. Some stopsword will be also removed.

        Args:
            filename (str): name of the dataset

        Returns:
            pd.DataFrame: Pandas DataFrame with two columns(label, text)
        """

        # import data
        dataframe = pd.read_csv(self.path + filename, sep="\t", header=None)

        # stops that will remain in the corpus
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

```
34          relevant_stops = set(['nicht', 'nichts', 'kein','kein', 'keine', 'keinem', '
                keinen', 'keiner', 'keines'])
35
36          # stops to remove
37          stops = set(stopwords.words('german')) - relevant_stops
38
39
40
41          # process data to create corpus
42          label_col, text_col = (dataframe[1].tolist(), dataframe[3].tolist())
43          binary_label_col, normalized_text_col = ([], [])
44
45          for index in range(len(label_col)-1):
46
47              label, text = (label_col[index], text_col[index])
48
49              # covert the labels into binary values
50              # 'gut' = 1, 'schlecht' = -1
51              if label == 'gut':
52                  binary_label_col.append(1)
53              else:
54                  binary_label_col.append(-1)
55
56
57              # text normalization
58              normalized_text = word_tokenize(str(text).lower())
59
60              # remove stops
61              for token in normalized_text:
62                  if token in stops: normalized_text.remove(token)
63
64              normalized_text_col.append(word_tokenize(str(text).lower()))
65
66
67          # create the corpus's data-structure
68          corpus = {'label': binary_label_col , 'text': normalized_text_col}
69
70          return pd.DataFrame(corpus)
71
72
73      def calc_term_weights_dictionary(self) ->dict:
74          """ Calculate the weight of each term and store them in a dictionary.
75          the weight is calulated with term-frequency * label.
76
77          Returns:
78              dict: store the weight as a value of a term
79          """
80          term_weights_dict = {}
81
82          for row in self.corpus.iterrows():
83              label, text = row[-1]
84
85              term_freq_dict = Counter(text)
86
87              for term in term_freq_dict:
88
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

```
                if term not in term_weights_dict:
                    term_weights_dict.update({ term: label * term_freq_dict[term] })
                else:
                    # sum the term fequencies
                    # if the class is 'schlecht' the term's frequency will be
                        negative
                    term_weights_dict[term] += label * term_freq_dict[term]

        return term_weights_dict









if __name__ == "__main__":
    classifier = SVM_Classifier()
    dictionary = classifier.term_weights_dictionary


    sorted_term_weight_list = sorted(dictionary.items(), key=itemgetter(1))

    print("-"*50, "\n Top 100 result for class 'gut': \n", "-"*50)
    for term, weight in sorted_term_weight_list[::-1][:100]:
        print(" ", weight, "\t", term)

    print("\n"*3)

    print("-"*50, "\n Top 100 result for class 'schlecht': \n", "-"*50)
    for term, weight in sorted_term_weight_list[:100]:
        print(" ", weight, "\t", term)
```

code/script.py

OUTPUT:

```
--------------------------------------------------
Top 100 result for class 'gut':
--------------------------------------------------
 24534          spiel
 22102          !
 19037          ist
 18307          es
 15812          cool
 14978          das
 13549          .
 12106          macht
 10989          :
 10854          super
 10760          ich
 10104          und
 10019          )
 9793   geil
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

```
    8686    aber
    8514    gut
    8346    sehr
    7933    einfach
    7425    man
    7323    spaß
    7083    ,
    6499    ein
    6411    nur
    6316    die
    4627    echt
    4427    –
    4038    finde
    3973    so
    3931    zu
    3847    noch
    3601    viel
    3489    nachbarn
    3476    auch
    3024    süchtig
    2906    kann
    2805    suche
    2802    für
    2692    richtig
    2586    der
    2569    mich
    2531    toll
    2478    wenn
    2464    ;
    2463    game
    2439    mit
    2390    gutes
    2365    dieses
    2252    voll
    2128    adden
    2127    spiele
    2058    liebe
    2051    cooles
    2023    top
    1967    hammer
    1934    tolles
    1900    bin
    1765    *
    1717    mir
    1686    neue
    1679    hat
    1668    geiles
    1647    beste
    1637    sind
    1623    sonst
    1616    immer
    1602    ...
    1541    weiter
    1531    klasse
    1471    muss
    1451    alles
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

```
  1431    langweilig
  1428    bitte
  1420    freunde
  1411    manchmal
  1394    wie
  1385    zeitvertreib
  1350    app
  1337    also
  1329    spass
  1298    ganz
  1271    gibt
  1250    lol
  1240    is
  1227    5
  1202    sterne
  1198    mega
  1189    wäre
  1176    was
  1153    eine
  1131    auf
  1114    spielen
  1087    läuft
  1062    könnte
  1057    empfehlen
  1057    dass
  1040    ihr
  1000    wirklich
  991     weil
  987     jeden
  985     nie



 ----------------------------------------------------
Top 100 result for class 'schlecht':
 ----------------------------------------------------
 -2016           nicht
 -1837           mehr
 -1183           seit
 -987    update
 -953    beheben
 -902    komme
 -721    rein
 -711    ?
 -711    ins
 -678    weg
 -641    event
 -632    war
 -619    stürzt
 -588    geht
 -564    dem
 -542    nichts
 -534    ständig
 -498    soll
 -473    keinen
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

```
130     -453    heute
131     -419    letzten
132     -408    verbindung
133     -407    gar
134     -367    neu
135     -359    komm
136     -355    ab
137     -353    server
138     -350    deinstalliert
139     -343    anfangen
140     -321    tagen
141     -318    fehler
142     -308    starten
143     -303    vorne
144     -303    scheiße
145     -302    scheiß
146     -299    nix
147     -279    gelöscht
148     -269    startet
149     -257    behebt
150     -255    lässt
151     -246    werde
152     -226    öffnen
153     -221    22
154     -218    wieder
155     -218    steht
156     -217    stern
157     -210    möglich
158     -208    obwohl
159     -202    nun
160     -198    mist
161     -191    dann
162     -190    support
163     -189    0
164     -176    gestern
165     -170    spielstand
166     -168    null
167     -165    %
168     -157    schlüssel
169     -154    lädt
170     -150    kotzen
171     -145    -.-
172     -143    wurde
173     -137    1
174     -137    gekauft
175     -136    anmelden
176     -136    zynga
177     -134    behoben
178     -133    enttäuscht
179     -132    sauer
180     -130    bekomme
181     -129    wollte
182     -127    raus
183     -125    investiert
184     -121    zurück
185     -121    passiert
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - st145113@stud.uni-stuttgart.de

```
186     -119    bildschirm
187     -118    beendet
188     -115    sofort
189     -115    deinstallieren
190     -114    abzocke
191     -113    belohnung
192     -113    scheisse
193     -112    installiert
194     -112    dauernd
195     -111    beim
196     -108    laden
197     -106    langsam
198     -106    installieren
199     -105    ärgerlich
200     -104    andauernd
201     -103    wochen
202     -99     seid
203     -98     trotz
204     -97     sekunden
205     -96     morgen
206     -95     zeigt
207     -94     angezeigt
208     -93     plötzlich
209     -91     bricht
210     -89     müll
```