EDA

December 16, 2019

0.1 1) PACKAGES AND LIBRARIES INSTALLATION

[1]:
```
!pip install 'plotnine[all]'
!pip install wordcloud
```

```
Requirement already satisfied: plotnine[all] in /usr/local/lib/python3.6/dist-
packages (0.5.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages
(from plotnine[all]) (1.17.4)
Requirement already satisfied: scipy>=1.0.0 in /usr/local/lib/python3.6/dist-
packages (from plotnine[all]) (1.3.3)
Requirement already satisfied: mizani>=0.5.2 in /usr/local/lib/python3.6/dist-
packages (from plotnine[all]) (0.5.4)
Requirement already satisfied: descartes>=1.1.0 in
/usr/local/lib/python3.6/dist-packages (from plotnine[all]) (1.1.0)
Requirement already satisfied: statsmodels>=0.8.0 in
/usr/local/lib/python3.6/dist-packages (from plotnine[all]) (0.10.2)
Requirement already satisfied: patsy>=0.4.1 in /usr/local/lib/python3.6/dist-
packages (from plotnine[all]) (0.5.1)
Requirement already satisfied: pandas>=0.23.4 in /usr/local/lib/python3.6/dist-
packages (from plotnine[all]) (0.25.3)
Requirement already satisfied: matplotlib>=3.0.0 in
/usr/local/lib/python3.6/dist-packages (from plotnine[all]) (3.1.2)
Collecting scikit-misc; extra == "all"
  Downloading https://files.pythonhosted.org/packages/52/12/5a3f2148c131b3
d9830ef3508e597cbfc790c3e704d7aa80cc1620485495/scikit_misc-0.1.1-cp36-cp36m-many
linux1_x86_64.whl (9.2MB)
     || 9.2MB 3.9MB/s
Requirement already satisfied: scikit-learn; extra == "all" in
/usr/local/lib/python3.6/dist-packages (from plotnine[all]) (0.21.3)
Requirement already satisfied: palettable in /usr/local/lib/python3.6/dist-
packages (from mizani>=0.5.2->plotnine[all]) (3.3.0)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages
(from patsy>=0.4.1->plotnine[all]) (1.12.0)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-
packages (from pandas>=0.23.4->plotnine[all]) (2018.9)
Requirement already satisfied: python-dateutil>=2.6.1 in
```

```
/usr/local/lib/python3.6/dist-packages (from pandas>=0.23.4->plotnine[all])
(2.6.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib>=3.0.0->plotnine[all])
(2.4.5)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib>=3.0.0->plotnine[all])
(1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-
packages (from matplotlib>=3.0.0->plotnine[all]) (0.10.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-
packages (from scikit-learn; extra == "all"->plotnine[all]) (0.14.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-
packages (from kiwisolver>=1.0.1->matplotlib>=3.0.0->plotnine[all]) (42.0.2)
Installing collected packages: scikit-misc
Successfully installed scikit-misc-0.1.1
Requirement already satisfied: wordcloud in /usr/local/lib/python3.6/dist-
packages (1.5.0)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.6/dist-
packages (from wordcloud) (1.17.4)
Requirement already satisfied: pillow in /usr/local/lib/python3.6/dist-packages
(from wordcloud) (4.3.0)
Requirement already satisfied: olefile in /usr/local/lib/python3.6/dist-packages
(from pillow->wordcloud) (0.46)
```

[0]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
import requests
from wordcloud import WordCloud,STOPWORDS
from plotnine import *
```

0.2 2) DATABASE

The main data was downloaded from Kaggle https://www.kaggle.com/crowdflower/twitter-airline-sentiment

0.2.1 2.1) Web scraping data

The idea here is to increase our dataset. We consider relevant to add the **total amount of followers** and if the **account is verified** or not for each twitter account of our database.

Please, find below a small sample.

[3]:
```
accounts = ('jnardino', 'cairdin', 'yvonnalynn')

for account in accounts:
    url = 'https://www.twitter.com/' + account
```

```
    r = requests.get(url)
    soup = BeautifulSoup(r.content, "lxml")

    f = soup.find('li', class_="ProfileNav-item--followers")
    title = f.find('a')['title']
    print(list((title, account)))
```

```
['7,727 Followers', 'jnardino']
['517 Followers', 'cairdin']
['1,821 Followers', 'yvonnalynn']
```

0.2.2 2.2) Loading of dataset

[0]:
```
# The new dataset
tweets = pd.read_csv("https://raw.githubusercontent.com/SarahBuechner/
 ↪DMML2019_Team_Google/master/data/Tweets.csv")
```

[5]:
```
# Dataframe tweets shape
print(tweets.shape)
# Data types
print(tweets.dtypes)
```

```
(12265, 18)
Unnamed: 0                      int64
tweet_id                        int64
airline_sentiment              object
airline_sentiment_confidence  float64
negativereason                 object
negativereason_confidence     float64
airline                        object
airline_sentiment_gold         object
name                           object
negativereason_gold            object
retweet_count                   int64
text                           object
tweet_coord                    object
tweet_created                  object
tweet_location                 object
user_timezone                  object
Followers                       int64
Verified                         bool
dtype: object
```

0.3 3) DATA CLEANING

Here we are elimintating all the data that is not usefull to our analysis. We erase the columns that we are not going to use. Then, we convert the column with the tweet creation time into a timestamp.

[0]:
```
# Drop the columns we don't use and converting tweet_created to datetime
try:
  tweets = tweets.drop(columns=['airline_sentiment_gold',␣
 ↪'negativereason_gold', 'tweet_coord', 'user_timezone'])
except KeyError:
  print("The columns have already been removed")
finally:
  tweets['tweet_created'] = pd.to_datetime(tweets['tweet_created'])
```

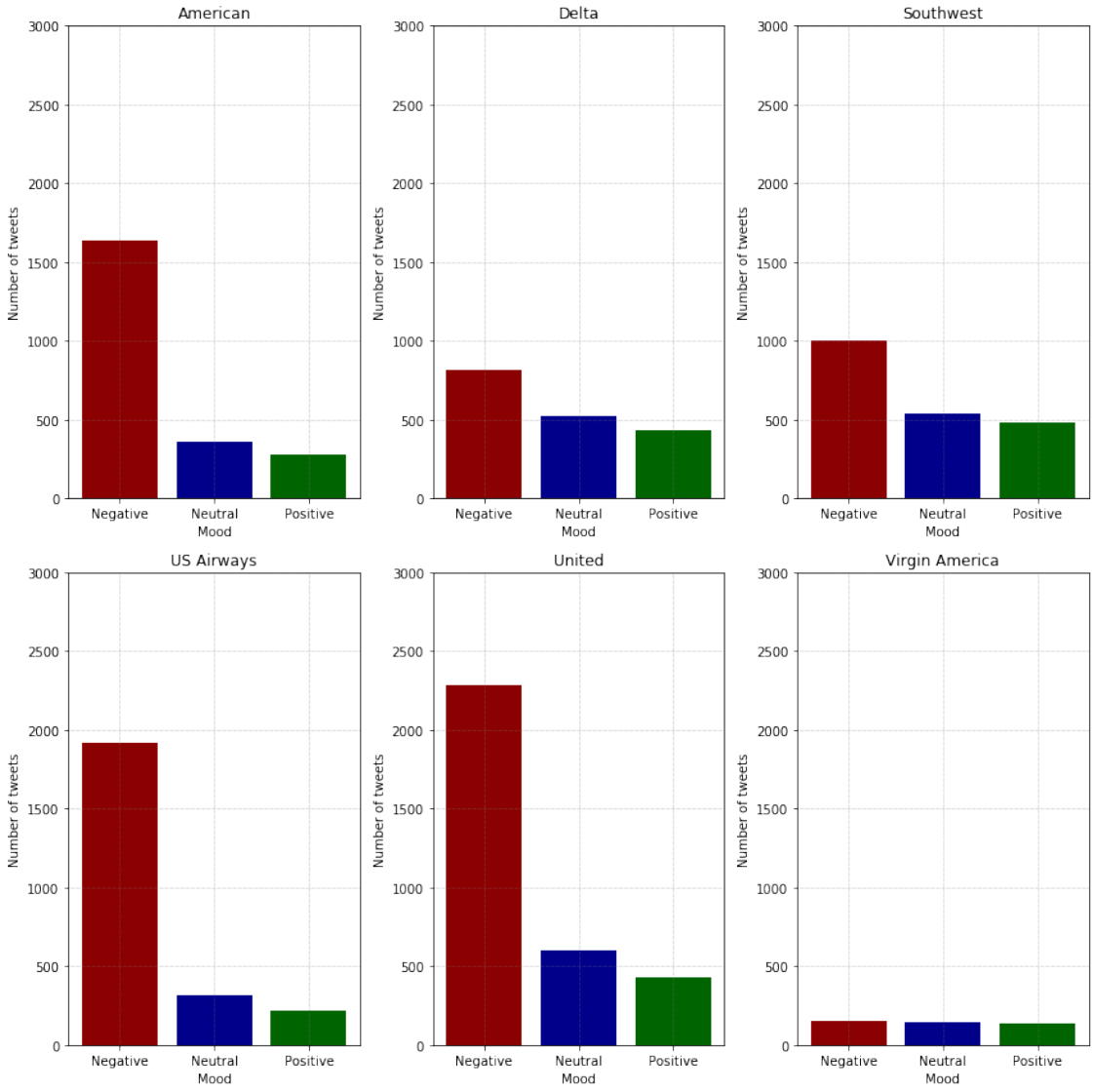
0.4 4) EXPLORATORY DATA ANALYSIS

0.4.1 4.1) Sentiment Analysis by Airline

We want to visualize the **distribution sentiment analysis by company**.

[7]:
```
def sentiment_by_airline(Airline):
  df=tweets[tweets['airline']==Airline]
  count=df['airline_sentiment'].value_counts()
  Index = [1,2,3]
  plt.bar(Index,count,color=['darkred', 'darkblue', 'darkgreen'])
  plt.xticks(Index,['Negative','Neutral','Positive'])
  plt.ylabel('Number of tweets')
  plt.xlabel('Mood')
  plt.title(Airline)
  plt.ylim(0,3000)
  plt.grid(which='major', linestyle=':', linewidth='0.5', color='grey')

airlines = ["American", "Delta", "Southwest", "US Airways", "United", "Virgin␣
 ↪America"]
plt.figure(1,figsize=(12, 12))


for airline in airlines:
  plt.subplot(231 + airlines.index(airline))
  sentiment_by_airline(airline)
plt.tight_layout()

count=tweets['airline_sentiment'].value_counts()
print("The most common class is negative mood = "+"{:.2%}".format(count[0]/
 ↪sum(count[0:3])))
```

```
The most common class is negative mood = 63.66%
```



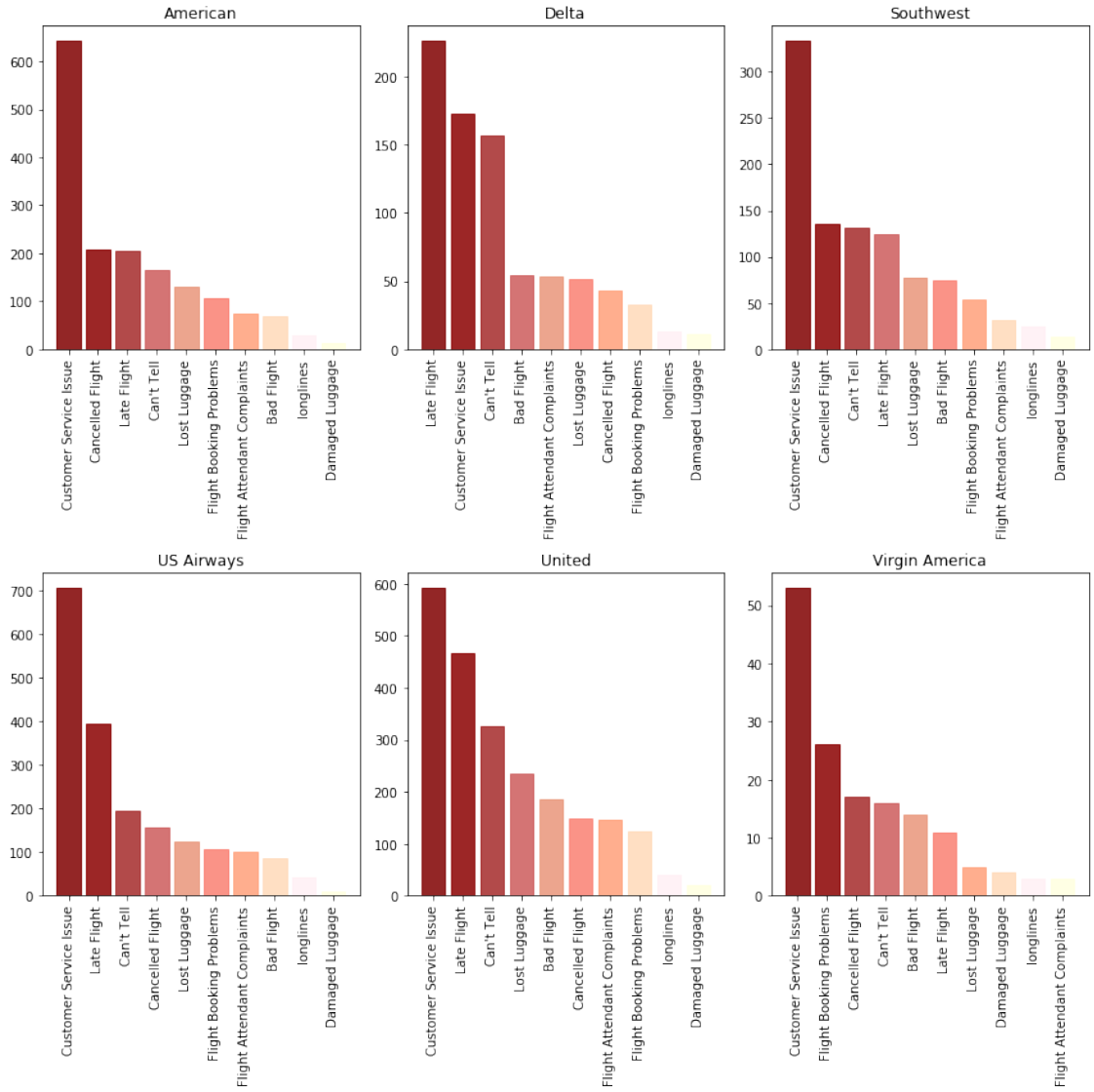
0.4.2 4.2) Negative Reason by Airline

In the previous graph we saw that the most common class was the *Negative mood*. Now we want to visualize the **negative ratings clustered on topics**.

[8]:
```
cmap = ['maroon', 'darkred', 'brown', 'indianred', 'darksalmon', 'salmon',␣
 ↪'lightsalmon', 'peachpuff', 'lavenderblush', 'lightyellow']
def negative_reason(Airline):
  df=tweets[tweets['airline']== Airline]
  Neg_reasons_values = df['negativereason'].value_counts(sort=True)
  Neg_reasons_labels = dict(df['negativereason'].value_counts(sort=True))
  Index = range(len(Neg_reasons_values))
  barplot = plt.bar(Index,Neg_reasons_values, alpha = 0.85)
```

```
  plt.xticks(Index,Neg_reasons_labels, rotation = 90)
  plt.title(Airline)

  for color in cmap:
    barplot[cmap.index(color)].set_color(color)

  return ("The 3 main negative topics of " + Airline + " are: (1) " +␣
 ↪list(Neg_reasons_labels.keys())[0] +
  ", (2) " + list(Neg_reasons_labels.keys())[1] + " and (3) " +␣
 ↪list(Neg_reasons_labels.keys())[2])


airlines = ["American", "Delta", "Southwest", "US Airways", "United", "Virgin␣
 ↪America"]
plt.figure(1,figsize=(12, 12))

for airline in airlines:
  plt.subplot(231 + airlines.index(airline))
  print(negative_reason(airline))

plt.tight_layout()
```

```
The 3 main negative topics of American are: (1) Customer Service Issue, (2)
Cancelled Flight and (3) Late Flight
The 3 main negative topics of Delta are: (1) Late Flight, (2) Customer Service
Issue and (3) Can't Tell
The 3 main negative topics of Southwest are: (1) Customer Service Issue, (2)
Cancelled Flight and (3) Can't Tell
The 3 main negative topics of US Airways are: (1) Customer Service Issue, (2)
Late Flight and (3) Can't Tell
The 3 main negative topics of United are: (1) Customer Service Issue, (2) Late
Flight and (3) Can't Tell
The 3 main negative topics of Virgin America are: (1) Customer Service Issue,
(2) Flight Booking Problems and (3) Cancelled Flight
```

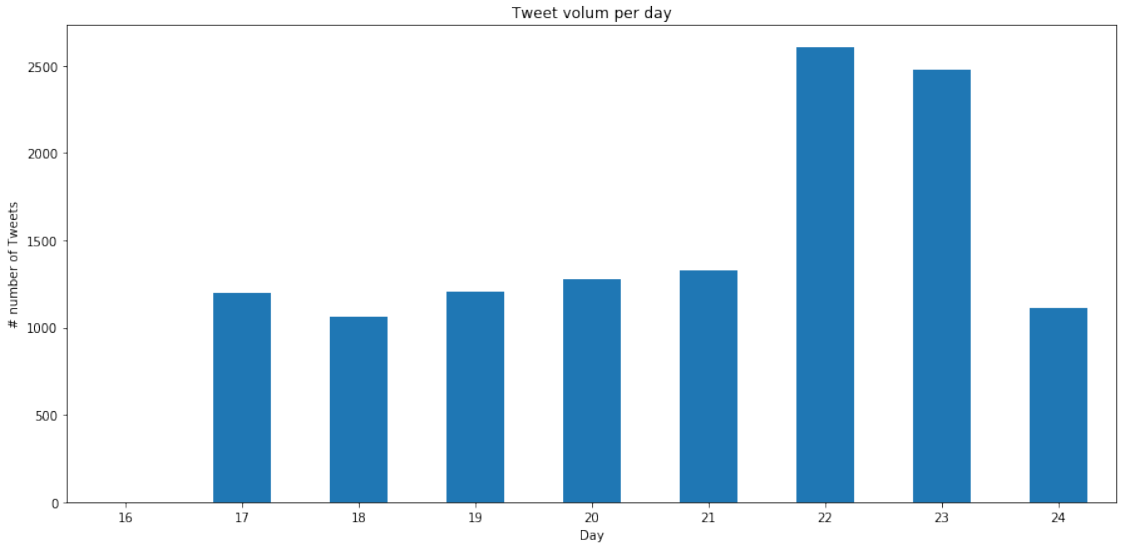


0.4.3 4.3) Tweet volum per day

Here we want to show the tweet distribution with regards to the day of publication.

[9]:
```
X = tweets['tweet_created']

fig, ax = plt.subplots(figsize=(15,7))
X.groupby([X.dt.day]).count().plot(kind="bar")
ax.set_title('Tweet volum per day')
ax.set_ylabel("# number of Tweets")
ax.set_xlabel("Day")
plt.xticks(rotation = 0)
```

[9]:
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8]), <a list of 9 Text xticklabel objects>)
```



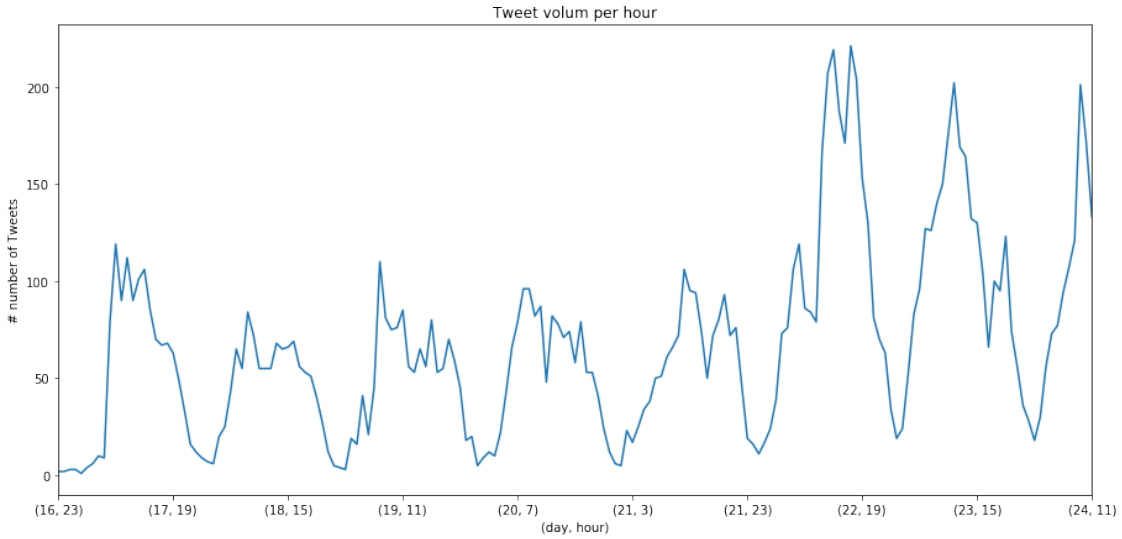
0.4.4 4.4) Tweet volum per hour and per day

Below you can see the tweet distribution on an hourly and daily basis.

[10]:
```
df = tweets[['tweet_created', 'airline_sentiment_confidence']]
df['tweet_created'] = pd.to_datetime(df['tweet_created'])
X = df['tweet_created']

fig, ax = plt.subplots(figsize=(15,7))
X.groupby([X.dt.day,X.dt.hour]).count().plot(ax=ax)
ax.set_title('Tweet volum per hour')
ax.set_xlabel("(day, hour)")
ax.set_ylabel("# number of Tweets")
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[10]:
```
Text(0, 0.5, '# number of Tweets')
```



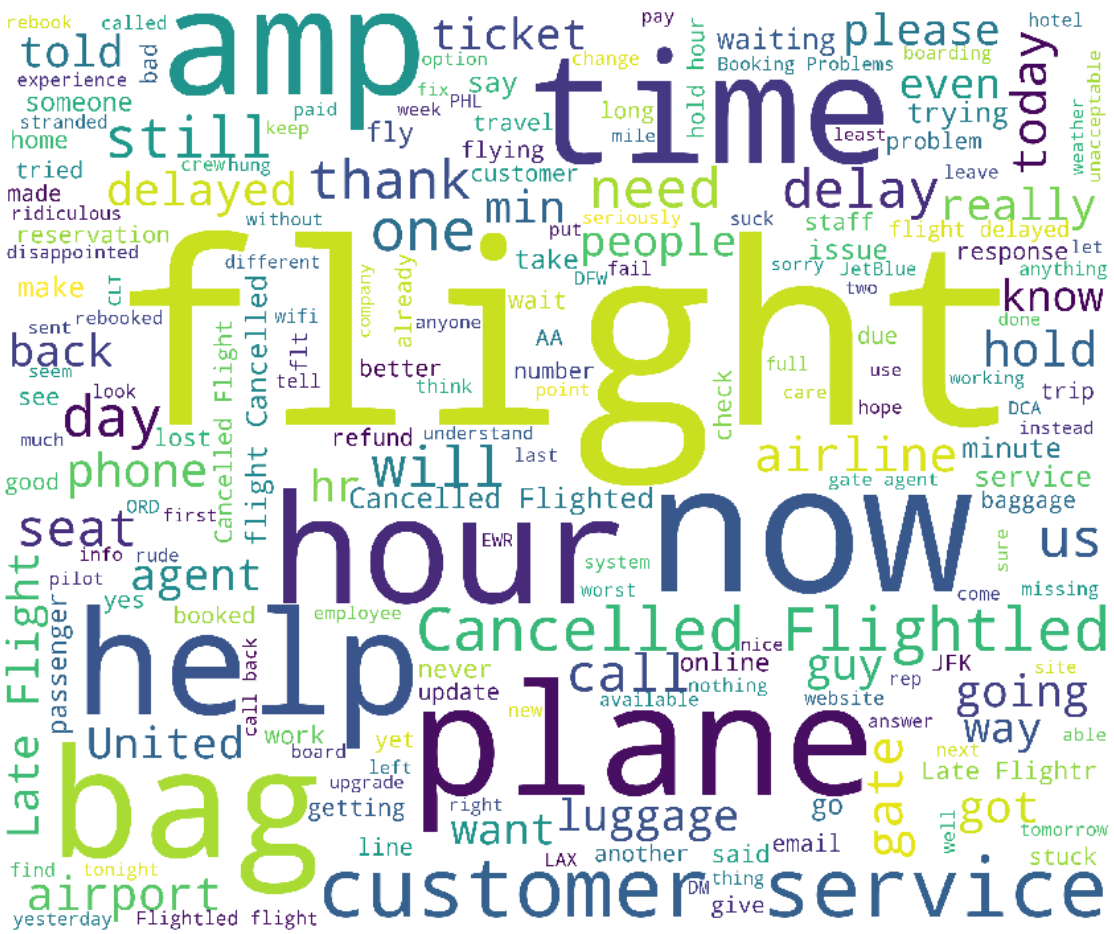
0.4.5 4.7) World cloud

4.7.1) Draw wordcloud function The `draw_wordcloud` function allows to plot the drawcloud according to the sentiment analysis:

Arguments: * `sentiment`: string representing the sentiment. The possible values are: `('positive', 'negative', 'neutral')`.

[0]:
```
def draw_wordcloud(sentiment):
    sentiment_tweets  =tweets[tweets['airline_sentiment']== sentiment]
    words = ' '.join(sentiment_tweets['text'])
    cleaned_word = " ".join([word for word in words.split() if 'http' not in␣
↪word and not word.startswith('@') and word != 'RT'])
    wordcloud =␣
↪WordCloud(stopwords=STOPWORDS,background_color='white',width=3000,␣
↪height=2500).generate(cleaned_word)
    plt.figure(1,figsize=(15,15))
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.show()
```

[12]:
```
draw_wordcloud('negative')
```



[13]:
```
draw_wordcloud('positive')
```

[14]:
```
draw_wordcloud('neutral')
```


please
JetBlue
fleet
gate
flying
fly
help
change
seat
will
need
flight
way
tomorrow
plane
got
amp
today
Cancelled Flightled
travel
see
go
bag
reservation
know
make
DM
now
check
airport
Thank
ticket
one
us
take
still
fleek
time
back
fleet
Booking Problems
DestinationDragon
Flight Booking
United
Chicago
Dallas
Nashville
Denver
Atlanta
Southwest
Vegas
Cancelled Flight
customer service
fleek JetBlue
Booking Problems