week-2

April 4, 2024

[127]:
```
import pandas as pd
import numpy as np
from tqdm import tqdm
from sklearn.svm import SVC
from sklearn import preprocessing, decomposition, model_selection, metrics,␣
 ↪pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from nltk import word_tokenize
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
```

[65]:
```
df=pd.read_csv('C:\\Users\\deeks\\Downloads\\bengaluru_house_prices.csv')
df.head()
```

[65]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

[66]:
```
df.shape
```

[66]:
```
(13320, 9)
```

[67]:
```
df.isnull()
```

[67]:
```
       area_type  availability  location   size  society  total_sqft   bath  \
0          False         False     False  False    False       False  False
1          False         False     False  False    False       False  False
2          False         False     False  False     True       False  False
3          False         False     False  False    False       False  False
4          False         False     False  False     True       False  False
…            …             …         …      …        …           …      …
13315      False         False     False  False    False       False  False
13316      False         False     False  False     True       False  False
13317      False         False     False  False    False       False  False
13318      False         False     False  False    False       False  False
13319      False         False     False  False     True       False  False

       balcony  price
0        False  False
1        False  False
2        False  False
3        False  False
4        False  False
…          …      …
13315    False  False
13316     True  False
13317    False  False
13318    False  False
13319    False  False

[13320 rows x 9 columns]
```

[68]:
```
df.isnull().sum().sum()
```

[68]:
```
6201
```

filling null values

[69]:
```
df.fillna(value=0)
```

[69]:
```
                  area_type   availability                  location  \
0      Super built-up  Area         19-Dec  Electronic City Phase II
1                Plot  Area  Ready To Move          Chikka Tirupathi
2             Built-up  Area  Ready To Move               Uttarahalli
3      Super built-up  Area  Ready To Move        Lingadheeranahalli
4      Super built-up  Area  Ready To Move                  Kothanur
…                    …             …                         …
13315         Built-up  Area  Ready To Move                Whitefield
13316  Super built-up  Area  Ready To Move              Richards Town
```

```
13317        Built-up  Area  Ready To Move     Raja Rajeshwari Nagar
13318  Super built-up  Area         18-Jun           Padmanabhanagar
13319  Super built-up  Area  Ready To Move              Doddathoguru

             size  society total_sqft  bath  balcony   price
0          2 BHK  Coomee        1056   2.0      1.0   39.07
1      4 Bedroom  Theanmp       2600   5.0      3.0  120.00
2          3 BHK        0       1440   2.0      3.0   62.00
3          3 BHK  Soiewre       1521   3.0      1.0   95.00
4          2 BHK        0       1200   2.0      1.0   51.00
…            …       …         …    …       …     …
13315  5 Bedroom  ArsiaEx       3453   4.0      0.0  231.00
13316      4 BHK        0       3600   5.0      0.0  400.00
13317      2 BHK  Mahla T       1141   2.0      1.0   60.00
13318      4 BHK  SollyCl       4689   4.0      1.0  488.00
13319      1 BHK        0        550   1.0      1.0   17.00

[13320 rows x 9 columns]
```

[70]:
```
df
```

[70]:
```
                 area_type   availability                  location  \
0      Super built-up  Area         19-Dec  Electronic City Phase II
1                Plot  Area  Ready To Move          Chikka Tirupathi
2           Built-up  Area  Ready To Move               Uttarahalli
3      Super built-up  Area  Ready To Move        Lingadheeranahalli
4      Super built-up  Area  Ready To Move                  Kothanur
…                    …            …                      …
13315       Built-up  Area  Ready To Move                Whitefield
13316  Super built-up  Area  Ready To Move              Richards Town
13317       Built-up  Area  Ready To Move     Raja Rajeshwari Nagar
13318  Super built-up  Area         18-Jun           Padmanabhanagar
13319  Super built-up  Area  Ready To Move              Doddathoguru

             size  society total_sqft  bath  balcony   price
0          2 BHK  Coomee        1056   2.0      1.0   39.07
1      4 Bedroom  Theanmp       2600   5.0      3.0  120.00
2          3 BHK      NaN       1440   2.0      3.0   62.00
3          3 BHK  Soiewre       1521   3.0      1.0   95.00
4          2 BHK      NaN       1200   2.0      1.0   51.00
…            …       …         …    …       …     …
13315  5 Bedroom  ArsiaEx       3453   4.0      0.0  231.00
13316      4 BHK      NaN       3600   5.0      NaN  400.00
13317      2 BHK  Mahla T       1141   2.0      1.0   60.00
13318      4 BHK  SollyCl       4689   4.0      1.0  488.00
13319      1 BHK      NaN        550   1.0      1.0   17.00
```

```
[13320 rows x 9 columns]
```

[71]:
```
df.describe()
```

[71]:
```
                bath       balcony         price
count  13247.000000  12711.000000  13320.000000
mean       2.692610      1.584376    112.565627
std        1.341458      0.817263    148.971674
min        1.000000      0.000000      8.000000
25%        2.000000      1.000000     50.000000
50%        2.000000      2.000000     72.000000
75%        3.000000      2.000000    120.000000
max       40.000000      3.000000   3600.000000
```

[72]:
```
df.dtypes
```

[72]:
```
area_type        object
availability     object
location         object
size             object
society          object
total_sqft       object
bath            float64
balcony         float64
price           float64
dtype: object
```

[73]:
```
df[df['price']>500]
```

[73]:
```
                  area_type   availability                location       size  \
7      Super built-up  Area  Ready To Move            Rajaji Nagar      4 BHK
62                Plot  Area  Ready To Move              Whitefield  4 Bedroom
159               Plot  Area  Ready To Move      Mahalakshmi Layout  4 Bedroom
408    Super built-up  Area         19-Jan            Rajaji Nagar      7 BHK
440               Plot  Area  Ready To Move              Whitefield  4 Bedroom
…                    …            …                       …        …
13095  Super built-up  Area  Ready To Move       Sathya Sai Layout      4 BHK
13104        Built-up  Area         19-Dec           Church Street      4 BHK
13119             Plot  Area  Ready To Move       Sathya Sai Layout  4 Bedroom
13197             Plot  Area  Ready To Move  Ramakrishnappa Layout  4 Bedroom
13200             Plot  Area  Ready To Move          Defence Colony  6 Bedroom

       society total_sqft  bath  balcony   price
7      Brway G       3300   4.0      NaN   600.0
62     Chranya       5700   5.0      3.0   650.0
159        NaN       3750   4.0      0.0   760.0
408        NaN      12000   6.0      3.0  2200.0
```

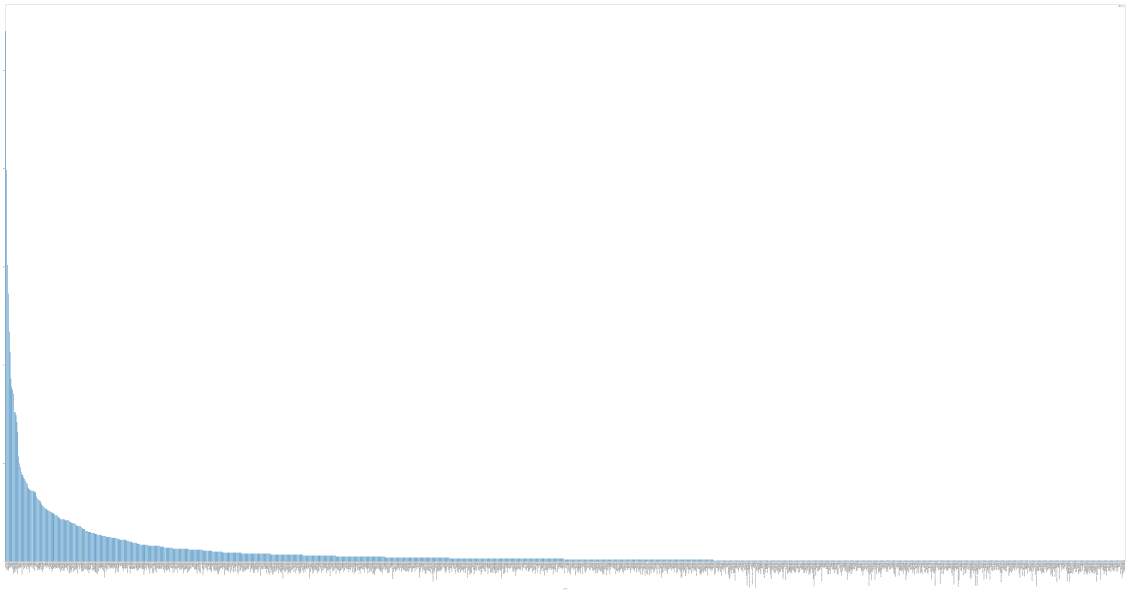
```
440         NaN      11890   4.0      3.0   700.0
…           …          …     …        …     …
13095   Prowshi       6652   6.0      1.0   660.0
13104   CoDast        2920   4.0      3.0   536.0
13119   Prowshi       6688   6.0      1.0   700.0
13197       NaN       9200   4.0      NaN  2600.0
13200       NaN       8000   6.0      3.0  2800.0

[241 rows x 9 columns]
```

[74]:
```
df['location'].value_counts()
```

[74]:
```
location
Whitefield                          540
Sarjapur  Road                      399
Electronic City                     302
Kanakpura Road                      273
Thanisandra                         234
                                   …
Bapuji Layout                         1
1st Stage Radha Krishna Layout        1
BEML Layout 5th stage                 1
singapura paradise                    1
Abshot Layout                         1
Name: count, Length: 1305, dtype: int64
```

[75]:
```
pd.DataFrame(df['location'].value_counts()).plot(kind='bar',figsize=[200,100])
```

[75]:
```
<Axes: xlabel='location'>
```



[76]:
```
import seaborn as sns
```

[77]:
```
sns.get_dataset_names()
```

[77]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic',
 'anagrams',
 'anagrams',
 'anscombe',
 'anscombe',
 'attention',
 'attention',
 'brain_networks',
 'brain_networks',
 'car_crashes',
 'car_crashes',
 'diamonds',
 'diamonds',
 'dots',
 'dots',
 'dowjones',
 'dowjones',
 'exercise',
 'exercise',
 'flights',
 'flights',
```

```
'fmri',
'fmri',
'geyser',
'geyser',
'glue',
'glue',
'healthexp',
'healthexp',
'iris',
'iris',
'mpg',
'mpg',
'penguins',
'penguins',
'planets',
'planets',
'seaice',
'seaice',
'taxis',
'taxis',
'tips',
'tips',
'titanic',
'titanic',
'anagrams',
'anscombe',
'attention',
'brain_networks',
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
'planets',
'seaice',
'taxis',
'tips',
'titanic']
```

[78]:
```
tips=sns.load_dataset("tips")
iris=sns.load_dataset("iris")
titanic=sns.load_dataset("titanic")
planets=sns.load_dataset("planets")
```

[79]:
```
tips
```

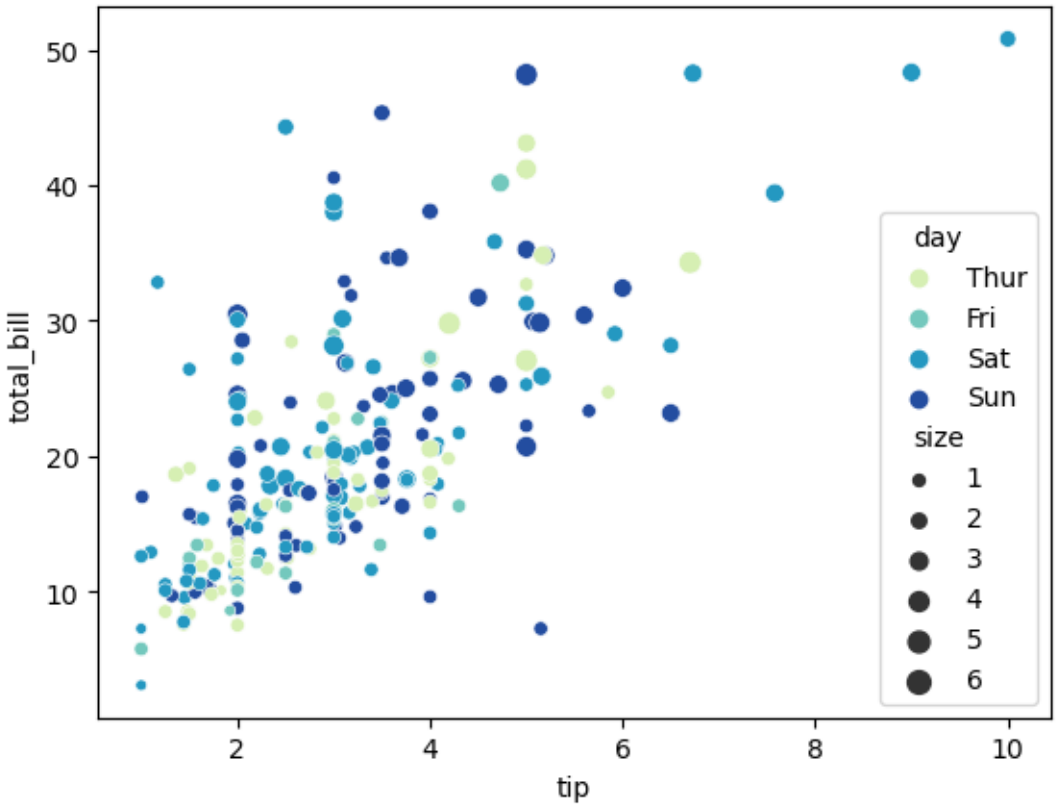
[79]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
..	…	…	…	…	…	…	…
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[244 rows x 7 columns]

[80]:
```
sns.
↪scatterplot(x="tip",y="total_bill",data=tips,hue="day",size="size",palette="YlGnBu")
```

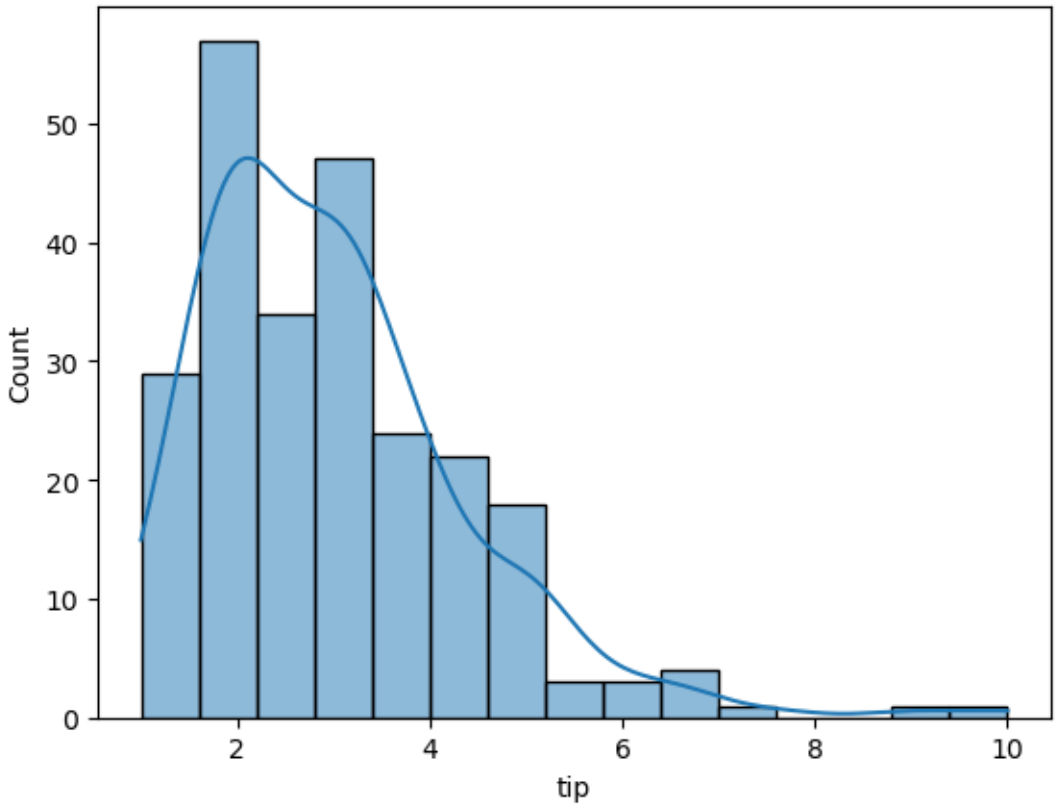
[80]:
```
<Axes: xlabel='tip', ylabel='total_bill'>
```



[81]:
```
sns.histplot(tips['tip'],kde=True,bins=15)
```

```
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

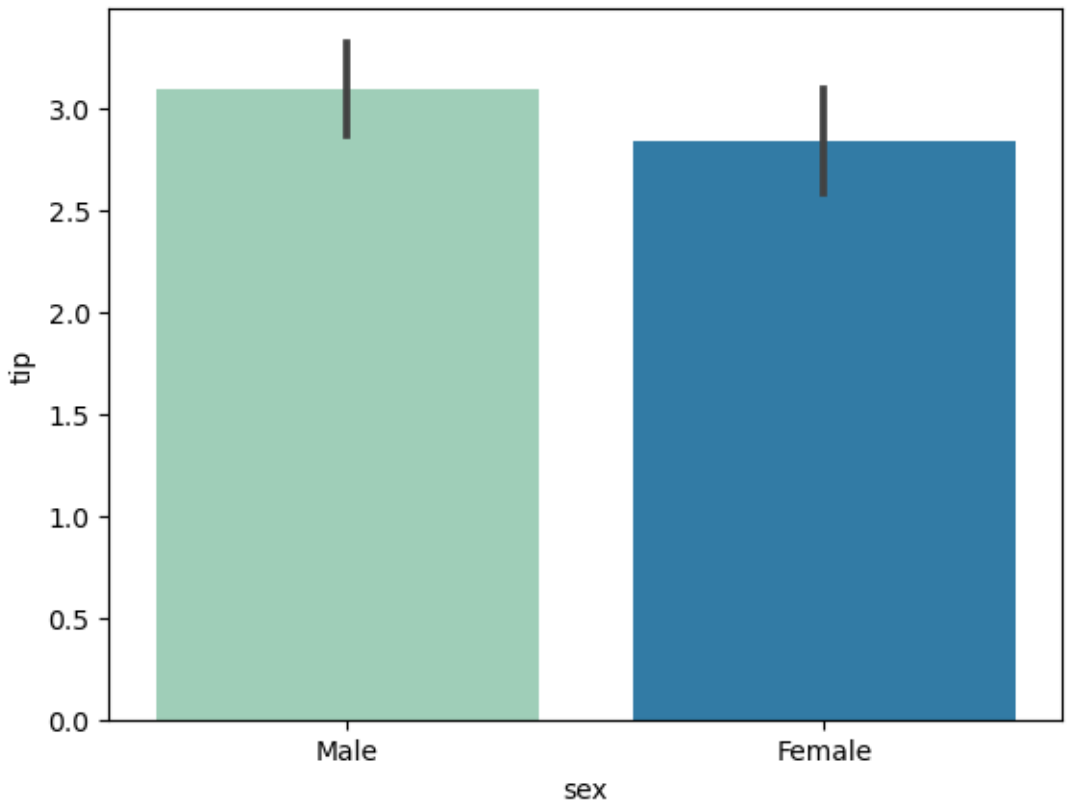
[81]:
```
<Axes: xlabel='tip', ylabel='Count'>
```



[82]:
```
sns.barplot(x="sex",y="tip",data=tips,palette="YlGnBu")
```

```
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

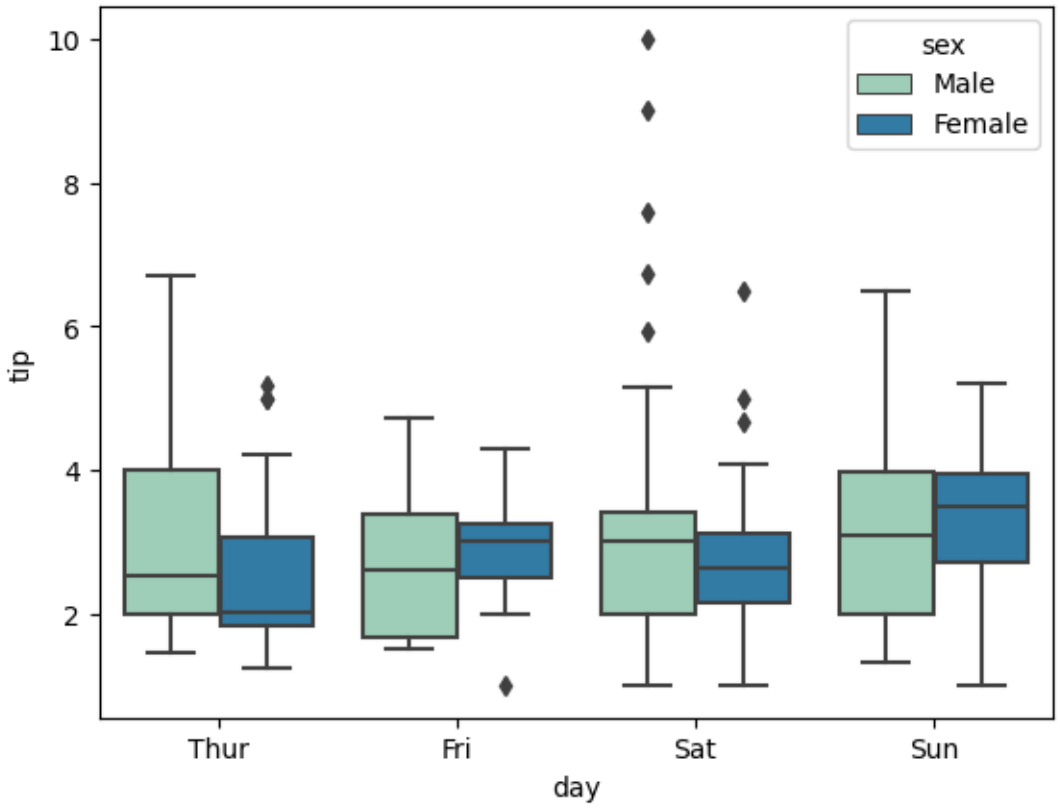
[82]:
```
<Axes: xlabel='sex', ylabel='tip'>
```



[83]:
```
sns.boxplot(x="day",y="tip",data=tips,hue="sex",palette="YlGnBu")
```

```
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

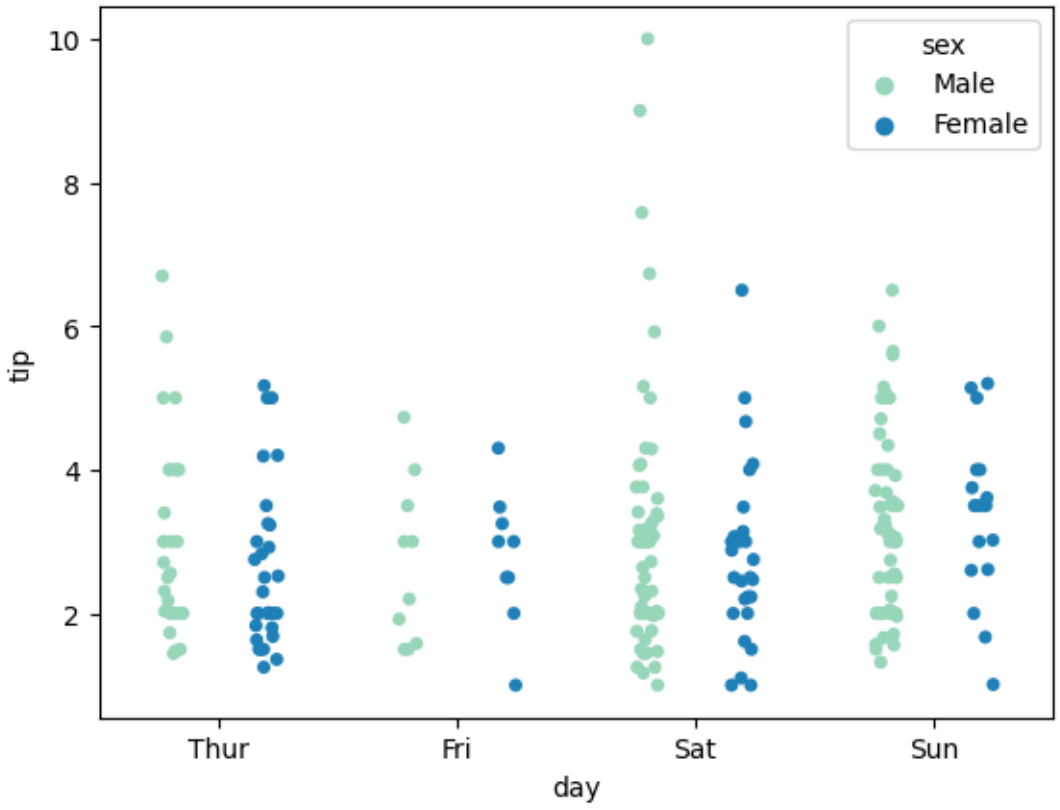
[83]:
```
<Axes: xlabel='day', ylabel='tip'>
```



[84]:
```
sns.stripplot(x="day",y="tip",data=tips,hue="sex",palette="YlGnBu",dodge=True)
```

```
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1057:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_data = data.groupby(
```

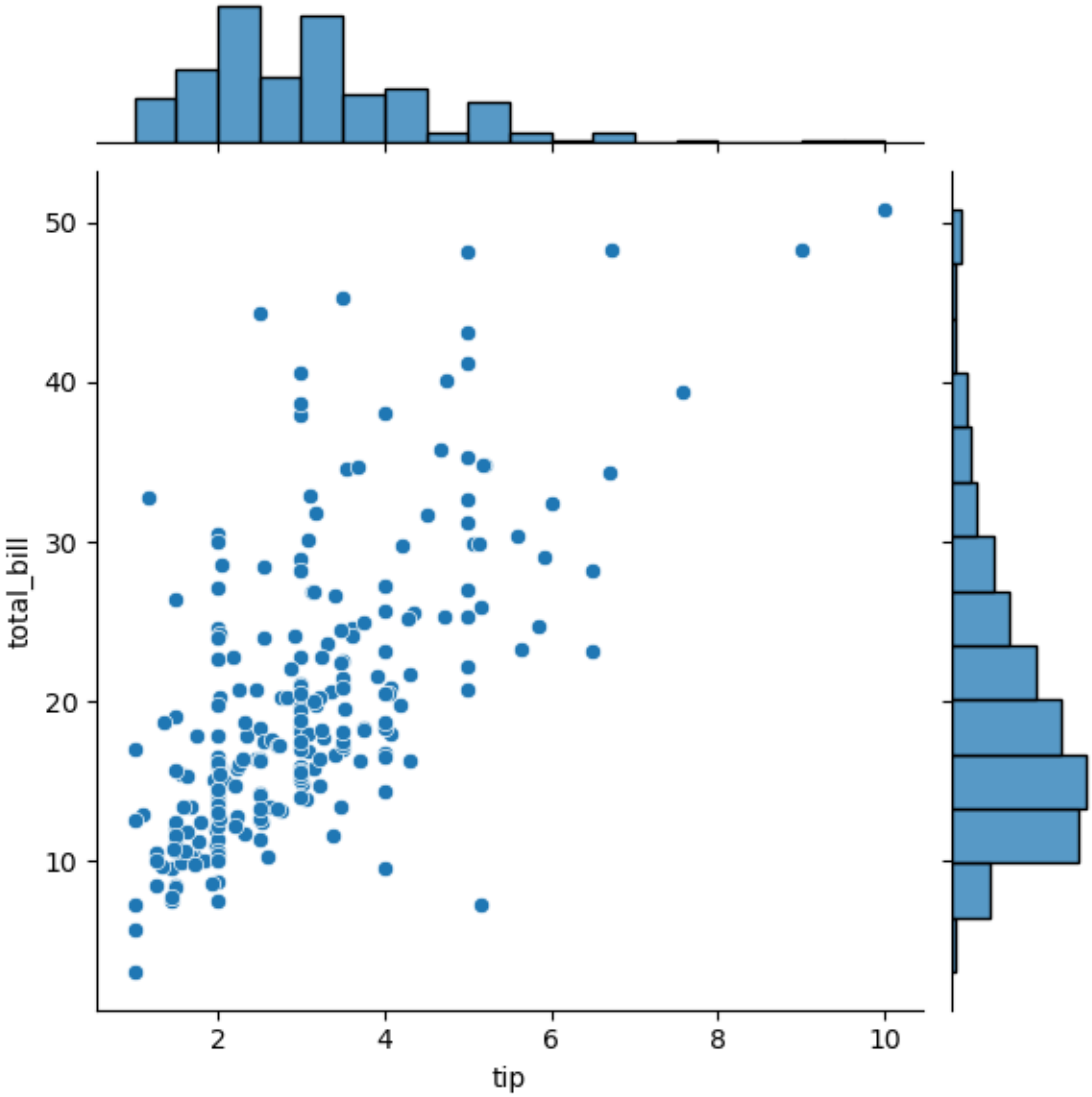
[84]:
```
<Axes: xlabel='day', ylabel='tip'>
```



[85]:
```
sns.jointplot(x="tip",y="total_bill",data=tips)
```

```
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[85]:
```
<seaborn.axisgrid.JointGrid at 0x1237c6fbd90>
```

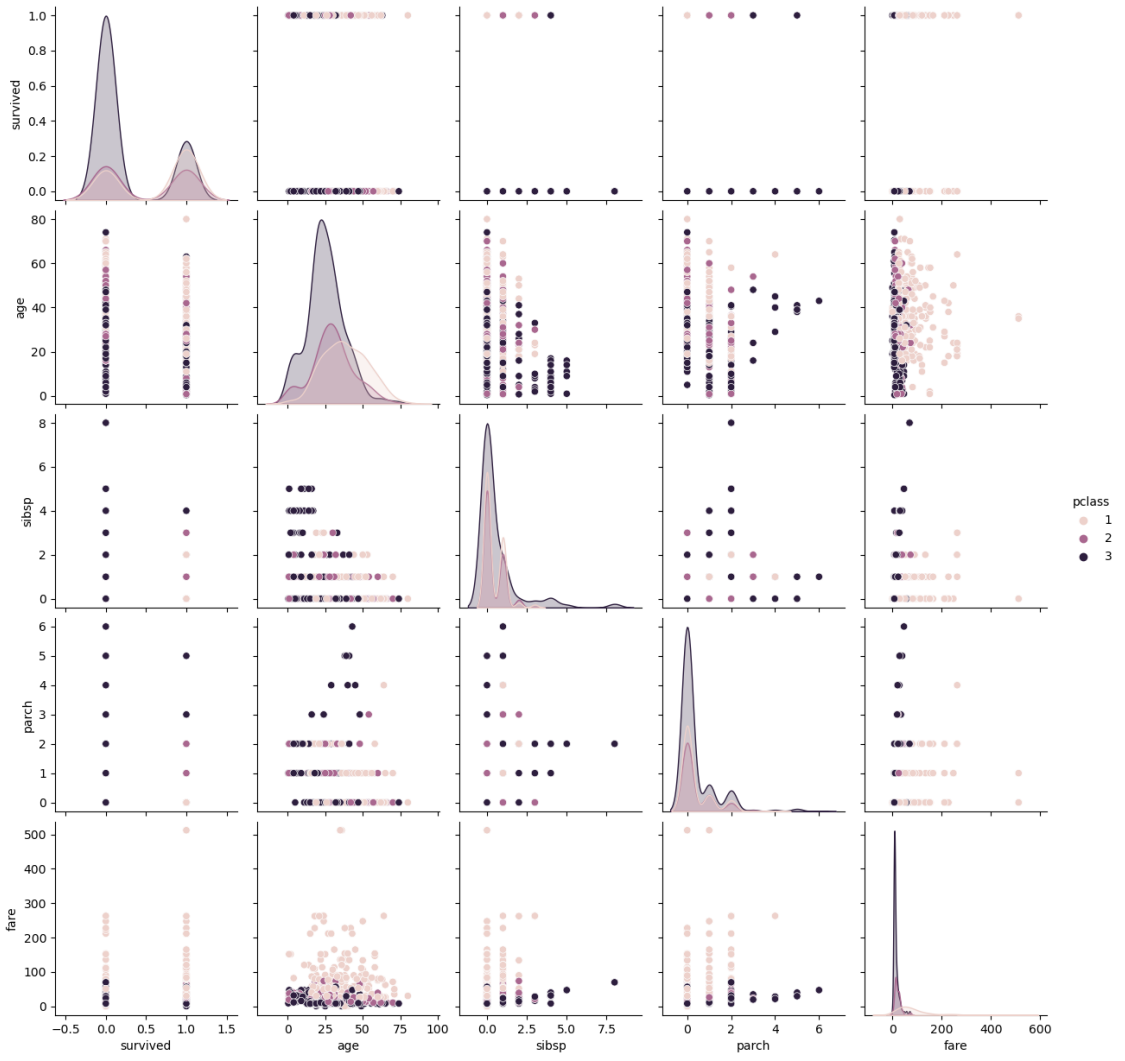


[86]:
```
sns.pairplot(titanic.select_dtypes(['number']),hue="pclass")
```

```
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
```

```
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\deeks\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[86]: `<seaborn.axisgrid.PairGrid at 0x1237ca44510>`



[87]:
```
df=pd.read_csv('C:\\Users\\deeks\\Downloads\\mushrooms.csv')
df.head()
```

[87]:
```
  class cap-shape cap-surface cap-color bruises odor gill-attachment  \
0     p         x           s         n       t    p               f
1     e         x           s         y       t    a               f
2     e         b           s         w       t    l               f
3     p         x           y         w       t    p               f
4     e         x           s         g       f    n               f

  gill-spacing gill-size gill-color  … stalk-surface-below-ring  \
0            c         n          k  …                        s
1            c         b          k  …                        s
2            c         b          n  …                        s
3            c         n          n  …                        s
4            w         b          k  …                        s

  stalk-color-above-ring stalk-color-below-ring veil-type veil-color  \
0                      w                      w         p          w
1                      w                      w         p          w
2                      w                      w         p          w
3                      w                      w         p          w
4                      w                      w         p          w

  ring-number ring-type spore-print-color population habitat
0           o         p                 k          s       u
1           o         p                 n          n       g
2           o         p                 n          n       m
3           o         p                 k          s       u
4           o         e                 n          a       g

[5 rows x 23 columns]
```

[88]:
```
import numpy as np
x=np.array(df["population"]).reshape(-1,1)
```

[89]:
```
x.shape
```

[89]:
```
(8124, 1)
```

[90]:
```
print(x)
```

```
[['s']
 ['n']
 ['n']
 …
 ['c']
 ['v']
 ['c']]
```

[91]:
```
y=np.array(df["habitat"])
```

[92]:
```
y.shape
```

[92]:
```
(8124,)
```

[93]:
```
print(y)
```

```
['u' 'g' 'm' … 'l' 'l' 'l']
```

[94]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x, y,test_size=0.30)
```

[95]:
```
x_train.shape
```

[95]:
```
(5686, 1)
```

[96]:
```
x_test.shape
```

[96]:
```
(2438, 1)
```

[97]:
```
y_train.shape
```

[97]:
```
(5686,)
```

[98]:
```
print(y_test)
```

```
['p' 'g' 'g' … 'd' 'g' 'p']
```

Logistic Regression

[99]:
```
from sklearn.linear_model import LogisticRegression
```

[100]:
```
lg=LogisticRegression()
```

[107]:
```
import pandas as pd
df = pd.read_csv('C:\\Users\\deeks\\Downloads\\spam.csv', encoding='latin1')
df = df.dropna(how="any", axis=1)
df.columns = ['target', 'message']

df.head()
```

[107]:

	target	message
0	ham	Go until jurong point, crazy.. Available only …
1	ham	Ok lar… Joking wif u oni…
2	spam	Free entry in 2 a wkly comp to win FA Cup fina…
3	ham	U dun say so early hor… U c already then say…
4	ham	Nah I don't think he goes to usf, he lives aro…

[113]:
```
import re
import string

def clean_text(text):
    '''Make text lowercase, remove text in square brackets, remove links,
    remove punctuation, and remove words containing numbers.'''
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

df['message_clean'] = df['message'].apply(clean_text)
df.head()
```

[113]:

	target	message	\
0	ham	Go until jurong point, crazy.. Available only …	
1	ham	Ok lar… Joking wif u oni…	
2	spam	Free entry in 2 a wkly comp to win FA Cup fina…	
3	ham	U dun say so early hor… U c already then say…	
4	ham	Nah I don't think he goes to usf, he lives aro…	

	message_clean
0	go until jurong point crazy available only in …
1	ok lar joking wif u oni
2	free entry in a wkly comp to win fa cup final…
3	u dun say so early hor u c already then say
4	nah i dont think he goes to usf he lives aroun…

[116]:
```
from nltk.corpus import stopwords
import string
import nltk
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))
more_stopwords = {'u', 'im', 'c'}
stop_words.update(more_stopwords)

def remove_stopwords(text):
    words = text.split()  # Tokenize the text into words
    cleaned_words = [word for word in words if word.lower() not in stop_words␣
 ↪and word not in string.punctuation]
    return ' '.join(cleaned_words)
```

```
df['message_clean'] = df['message_clean'].apply(remove_stopwords)
df.head()
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\deeks\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping corpora\stopwords.zip.
```

[116]:
```
  target                                            message  \
0    ham  Go until jurong point, crazy.. Available only …
1    ham                      Ok lar… Joking wif u oni…
2   spam  Free entry in 2 a wkly comp to win FA Cup fina…
3    ham  U dun say so early hor… U c already then say…
4    ham  Nah I don't think he goes to usf, he lives aro…

                                      message_clean
0  go jurong point crazy available bugis n great …
1                              ok lar joking wif oni
2  free entry wkly comp win fa cup final tkts may…
3                      dun say early hor already say
4        nah dont think goes usf lives around though
```

[119]:
```
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
def preprocess_data(text):
    # Clean puntuation, urls, and so on
    text = clean_text(text)

    text = ' '.join(word for word in text.split(' ') if word not in stop_words)

    text = ' '.join(stemmer.stem(word) for word in text.split(' '))

    return text
df['message_clean'] = df['message_clean'].apply(preprocess_data)
df.head()
```

[119]:
```
  target                                            message  \
0    ham  Go until jurong point, crazy.. Available only …
1    ham                      Ok lar… Joking wif u oni…
2   spam  Free entry in 2 a wkly comp to win FA Cup fina…
3    ham  U dun say so early hor… U c already then say…
4    ham  Nah I don't think he goes to usf, he lives aro…

                                      message_clean
0  go jurong point crazi avail bugi n great world…
1                                ok lar joke wif oni
2  free entri wkli comp win fa cup final tkt may …
3                      dun say earli hor alreadi say
```

```
4          nah dont think goe usf live around though
```

[136]:
```
train = pd.read_csv('C:\\Users\\deeks\\Downloads\\test.csv\\test.csv')
test = pd.read_csv('C:\\Users\\deeks\\Downloads\\train.csv\\train.csv')
sample = pd.read_csv('C:\\Users\\deeks\\Downloads\\sample_submission.csv')
```

[137]:
```
def multiclass_logloss(actual, predicted, eps=1e-15):
    """Multi class version of Logarithmic Loss metric.
    :param actual: Array containing the actual target classes
    :param predicted: Matrix with class predictions, one probability per class
    """
    # Convert 'actual' to a binary array if it's not already:
    if len(actual.shape) == 1:
        actual2 = np.zeros((actual.shape[0], predicted.shape[1]))
        for i, val in enumerate(actual):
            actual2[i, val] = 1
        actual = actual2

    clip = np.clip(predicted, eps, 1 - eps)
    rows = actual.shape[0]
    vsota = np.sum(actual * np.log(clip))
    return -1.0 / rows * vsota
```

[ ]:
```
from sklearn import preprocessing
lbl_enc = preprocessing.LabelEncoder()
y = lbl_enc.fit_transform(train.author.values)
```

[ ]:
```
tfv = TfidfVectorizer(min_df=3,  max_features=None,
            strip_accents='unicode', analyzer='word',token_pattern=r'\w{1,}',
            ngram_range=(1, 3), use_idf=1,smooth_idf=1,sublinear_tf=1,
            stop_words = 'english')

# Fitting TF-IDF to both training and test sets (semi-supervised learning)
tfv.fit(list(xtrain) + list(xvalid))
xtrain_tfv =  tfv.transform(xtrain)
xvalid_tfv = tfv.transform(xvalid)
```

[ ]:
```
clf = LogisticRegression(C=1.0)
clf.fit(xtrain_tfv, ytrain)
predictions = clf.predict_proba(xvalid_tfv)

print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
```

[ ]:
```
clf = MultinomialNB()
clf.fit(xtrain_tfv, ytrain)
predictions = clf.predict_proba(xvalid_tfv)
```

```
print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
```

[ ]:
```
svd = decomposition.TruncatedSVD(n_components=120)
svd.fit(xtrain_tfv)
xtrain_svd = svd.transform(xtrain_tfv)
xvalid_svd = svd.transform(xvalid_tfv)

scl = preprocessing.StandardScaler()
scl.fit(xtrain_svd)
xtrain_svd_scl = scl.transform(xtrain_svd)
xvalid_svd_scl = scl.transform(xvalid_svd)
```

[ ]:
```
clf = SVC(C=1.0, probability=True)
clf.fit(xtrain_svd_scl, ytrain)
predictions = clf.predict_proba(xvalid_svd_scl)

print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
```