Other Dosage Forms

In [1]:
```
"""
SELECT
ag_custom_case_number__c,
prod.name,
dose_form.name,
ROW_NUMBER() OVER(PARTITION BY ca.ag_custom_case_number__c, pic.name ORDER BY contentv
pic.name as issue_type,
convert_from(decode(contentver.versiondata, 'base64'),'UTF8') AS notes
FROM
bct_schema."CASE" ca inner join bct_schema."CASE__HISTORY" casehistory on ca.id = case
inner join bct_schema."CONTENTDOCUMENTLINK" contdoclink on ca.parentid = contdoclink.l
inner join bct_schema."CONTENTVERSION" contentver on contdoclink.contentdocumentid = c
inner join bct_schema."AG_PRODUCT__C" prod on ca.ag_product__c = prod.id
inner join bct_schema."AG_CASE_PRODUCT__C" cprod on ca.parentid = cprod.ag_case__c
inner join bct_schema."AG_DOSAGE_FORM__C" dose_form on cprod.ag_dosage_form__c = dose_
inner join bct_schema."AG_PCM_ISSUE__C" pi on ca.ag_custom_case_number__c  = pi.ag_pcm
inner join bct_schema."AG_PCM_ISSUE_CODE__C" pic on pic.id = pi.ag_as_reported_code__c
inner join bct_schema."AG_PCM_ISSUE_CODE_FAMILY__C" picf on pi.ag_cause_code_family__c
WHERE
dose_form.name IN ('Solution for injection in pre-filled pen','Vial - liquid','Vial -
ca.ag_intake_channel_type__c IS NOT NULL and
casehistory.field = 'Status' and
casehistory.newvalue = 'Intake Complete' and
contentver.src_createddate <= casehistory.src_createddate;
"""
```

Out[1]:
```
'\nSELECT\nag_custom_case_number__c,\nprod.name,\ndose_form.name,\nROW_NUMBER() OVER
(PARTITION BY ca.ag_custom_case_number__c, pic.name ORDER BY contentver.src_createdda
te DESC) AS notes_rank,\npic.name as issue_type,\nconvert_from(decode(contentver.vers
iondata, \'base64\'),\'UTF8\') AS notes\nFROM \nbct_schema."CASE" ca inner join bct_s
chema."CASE__HISTORY" casehistory on ca.id = casehistory.caseid \ninner join bct_sche
ma."CONTENTDOCUMENTLINK" contdoclink on ca.parentid = contdoclink.linkedentityid\ninn
er join bct_schema."CONTENTVERSION" contentver on contdoclink.contentdocumentid = con
tentver.contentdocumentid\ninner join bct_schema."AG_PRODUCT__C" prod on ca.ag_produc
t__c = prod.id\ninner join bct_schema."AG_CASE_PRODUCT__C" cprod on ca.parentid = cpr
od.ag_case__c\ninner join bct_schema."AG_DOSAGE_FORM__C" dose_form on cprod.ag_dosage
_form__c = dose_form.id\ninner join bct_schema."AG_PCM_ISSUE__C" pi on ca.ag_custom_c
ase_number__c  = pi.ag_pcm_sub_case_number_apex__c\ninner join bct_schema."AG_PCM_ISS
UE_CODE__C" pic on pic.id = pi.ag_as_reported_code__c\ninner join bct_schema."AG_PCM_
ISSUE_CODE_FAMILY__C" picf on pi.ag_cause_code_family__c = picf.id\nWHERE\ndose_form.
name IN (\'Solution for injection in pre-filled pen\',\'Vial - liquid\',\'Vial - lyop
hilized\',\'Software based device\',\'Tablet\') and \nca.ag_intake_channel_type__c IS
NOT NULL and\ncasehistory.field = \'Status\' and \ncasehistory.newvalue = \'Intake Co
mplete\' and \ncontentver.src_createddate <= casehistory.src_createddate; \n'
```

In [2]:
```
import sys
print(sys.executable)
print(sys.version)
print(sys.version_info)
```

```
C:\Users\gmodi\Anaconda3\envs\FastText\python.exe
3.7.13 (default, Mar 28 2022, 08:03:21) [MSC v.1916 64 bit (AMD64)]
sys.version_info(major=3, minor=7, micro=13, releaselevel='final', serial=0)
```

In [3]:
```
import numpy as np
import pandas as pd
import texthero as hero
from texthero import stopwords
from texthero import preprocessing
from texthero import visualization
from texthero import representation

from bs4 import BeautifulSoup

import fasttext
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import remove_stopwords

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn import preprocessing
```

In [10]:

```
masterData = pd.read_csv("C:/Users/gmodi/Downloads/data-1666644445166.csv")
#masterData = pd.read_csv("C:/Users/gmodi/Downloads/data-1671237323737.csv")
#masterData["issue_type"] = masterData["issue_type"].str.replace(' ', '_').replace('/
masterData["issue_type"] = masterData["issue_type"].str.replace(r'[^0-9a-zA-Z:,]+', '_
masterData["len"] = masterData["notes"].apply(len)
```

```
C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\ipykernel_launcher.py:4: Fut
ureWarning: The default value of regex will change from True to False in a future ver
sion.
  after removing the cwd from sys.path.
```

In [11]:

```
masterData.head(5)
```

Out[11]:

	ag_custom_case_number__c	product	name	notes_rank	issue_type	
0	19-0000093-PC-01	Aranesp	Solution for injection in pre-filled pen	1	autoinjector_activation_difficulty	<p>A reported
1	19-0000093-PC-01	Aranesp	Solution for injection in pre-filled pen	2	autoinjector_activation_difficulty	 Prod
2	19-0000745-PC-01	Enbrel	Solution for injection in pre-filled pen	1	drug_injection	<p>I <p>Date
3	19-0000745-PC-01	Enbrel	Solution for injection in pre-filled pen	2	drug_injection	
4	19-0000745-PC-01	Enbrel	Solution for injection in pre-filled pen	3	drug_injection	<p>Tr numbe

In [12]:

```
# Functions
```

In [13]:

```
def clean_notes(text):
    import re
    soup = BeautifulSoup(text, 'html.parser')
    list1 = [item.get_text() for item in list(soup.children)]
    list2 = [i for i in list1 if len(i) == max([len(i) for i in list1])]
    list3 = [re.sub('[^a-zA-Z:]+', ' ', _) for _ in list2]
    return list3[0]

def multiclass_roc_auc_score(y_test, y_pred, average="macro"):
    lb = preprocessing.LabelBinarizer()
    lb.fit(y_test)
    y_test = lb.transform(y_test)
    y_pred = lb.transform(y_pred)
    return roc_auc_score(y_test, y_pred, average=average)
```

In [14]:

```
def normalize(s):
    """
    Given a text, cleans and normalizes it. Feel free to add your own stuff.
    """
    s = s.lower()
    # Replace ips
    s = re.sub(r'\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}', ' _ip_ ', s)
    # Isolate punctuation
    s = re.sub(r'([.\(\)\!\?\-\\\/\,])', r' \1 ', s)
    # Remove some special characters
    s = re.sub(r'([\;\:\|•«\n])', ' ', s)
    # Replace numbers and symbols with language
    s = s.replace('&', ' and ')
    s = s.replace('@', ' at ')
    s = s.replace('0', ' zero ')
    s = s.replace('1', ' one ')
    s = s.replace('2', ' two ')
    s = s.replace('3', ' three ')
    s = s.replace('4', ' four ')
    s = s.replace('5', ' five ')
    s = s.replace('6', ' six ')
    s = s.replace('7', ' seven ')
    s = s.replace('8', ' eight ')
    s = s.replace('9', ' nine ')
    return s
```

Solution for injection in pre-filled pen

In [15]:

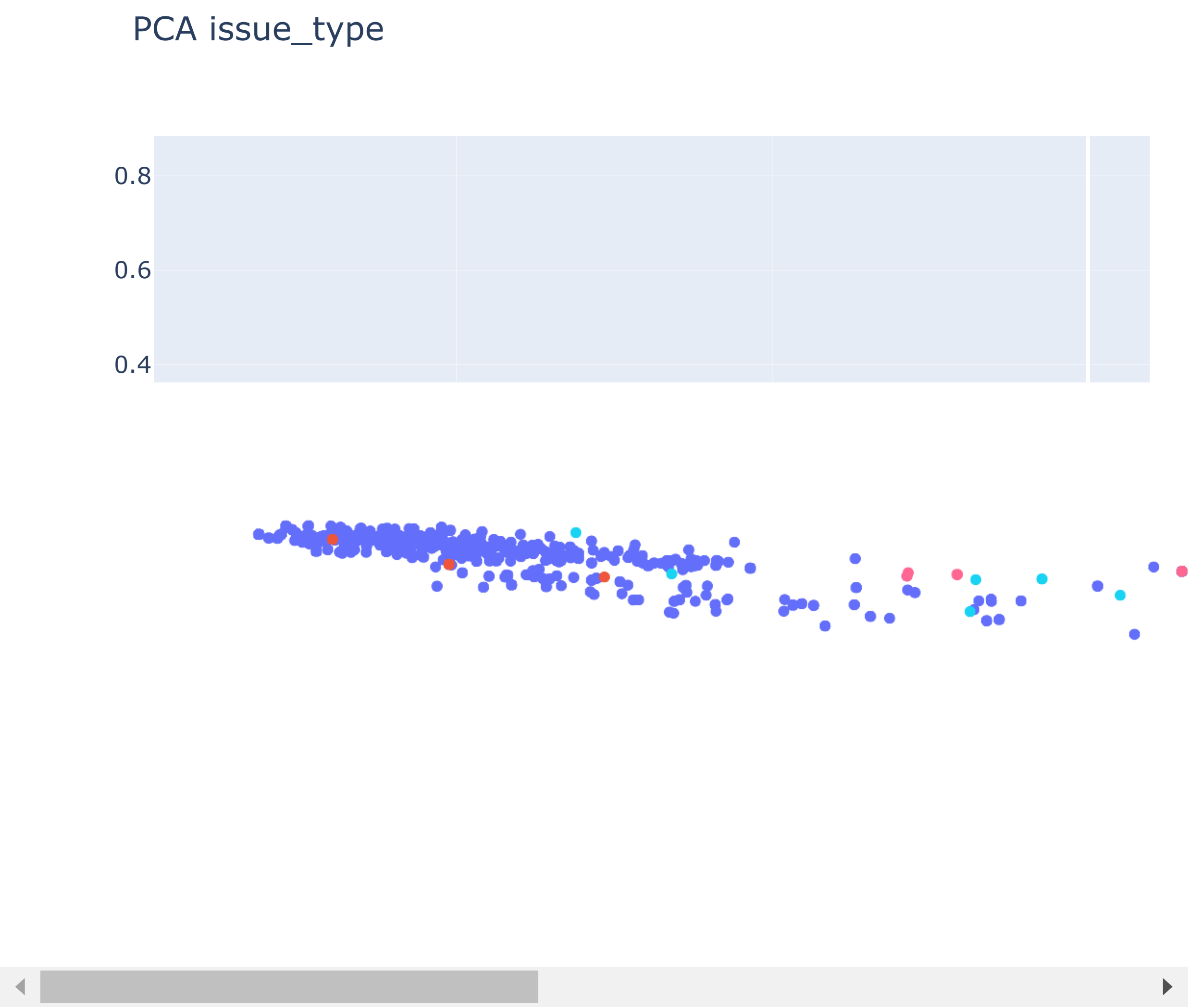
```
data = masterData.query(" name == 'Solution for injection in pre-filled pen' ").copy(
valueCount = data["issue_type"].value_counts(normalize=True).to_frame().cumsum()*100
data = data[data["issue_type"].isin(valueCount.index.tolist()[0:9])]
data = data[["notes","issue_type","len"]]

data['notes']=data['notes'].apply(lambda cw : clean_notes(cw))
data = data.query(" len > 500 ")

data["notes"] = data["notes"].apply(lambda x: ' '.join(simple_preprocess(x, min_len=4,
data["notes"] = data["notes"].apply(lambda x: remove_stopwords(''.join(x)))
```

In [16]:

```
data["pca"] = (data["notes"].pipe(representation.tfidf, max_features=100).pipe(represe
hero.scatterplot(data, col="pca", color="issue_type", title="PCA issue_type")
```



In [17]:

```
data["labeled_notes"] = data["issue_type"].apply(lambda x: '__label__' + x + " " ) + c

x_train,x_test,y_train,y_test = train_test_split(data[["labeled_notes","issue_type"]],
x_train.to_csv("C:/Users/gmodi/Downloads/x_train.csv",index=False,header=False)
x_test.to_csv("C:/Users/gmodi/Downloads/x_test.csv",index=False,header=False)

model = fasttext.train_supervised(input="C:/Users/gmodi/Downloads/x_train.csv", wordNg
model.test("C:/Users/gmodi/Downloads/x_test.csv",k=3)
```

Out[17]:

```
(1636, 0.33027709861450694, 0.9908312958435208)
```

In [18]:

```
# predict the data
x_test["predicted"] = x_test["labeled_notes"].apply(lambda x: model.predict(x)[0][0]).

#Create the confusion matrix
print(classification_report(x_test["issue_type"], x_test["predicted"]))
print(confusion_matrix(x_test["issue_type"], x_test["predicted"]))
multiclass_roc_auc_score(x_test["issue_type"], x_test["predicted"])
```

```
                                 precision    recall  f1-score   support

          Before_Activation_resolved       1.00      0.50      0.67        10
      activation_difficulty_resolved       0.98      0.95      0.97       172
autoinjector_activation_difficulty       0.97      0.96      0.96      1369
       autoinjector_user_mishandling       0.05      0.60      0.09         5
        carton_cosmetic_minor_damage       1.00      0.43      0.60         7
      carton_label_missing_incorrect       1.00      0.50      0.67         2
                   customer_feedback       1.00      0.71      0.83        17
                      drug_injection       0.93      0.77      0.84        52
                               other       1.00      1.00      1.00         2

                            accuracy                           0.94      1636
                           macro avg       0.88      0.71      0.74      1636
                        weighted avg       0.97      0.94      0.95      1636

[[   5    0    5    0    0    0    0    0    0]
 [   0  164    8    0    0    0    0    0    0]
 [   0    3 1309   54    0    0    0    3    0]
 [   0    0    2    3    0    0    0    0    0]
 [   0    0    4    0    3    0    0    0    0]
 [   0    0    1    0    0    1    0    0    0]
 [   0    0    5    0    0    0   12    0    0]
 [   0    0   10    2    0    0    0   40    0]
 [   0    0    0    0    0    0    0    0    2]]
```

Out[18]:

```
0.8468878491916008
```

In [19]:

```python
x_test["prediction"] = x_test["labeled_notes"].apply(lambda x: model.predict(x,3)).ast
#x_test["prediction"] = x_test["prediction"].astype(str)
#x_test["prediction"] = x_test["prediction"].str.replace('__label__','')

for i in range(len(x_test)):
    if x_test.issue_type.iloc[i] in x_test.prediction.iloc[i]: x_test.predicted.iloc[i
    else: x_test.predicted.iloc[i] = 0

#x_test.to_csv("C:/Users/gmodi/Downloads/x_test_results.csv")
x_test["predicted"].value_counts(normalize=True)*100
```

Out[19]:

```
1    95.904645
0     4.095355
Name: predicted, dtype: float64
```

In [20]:

```python
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_SIPFP.bin")
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_SIPFP.ftz")
```

Vial - liquid

In [21]:

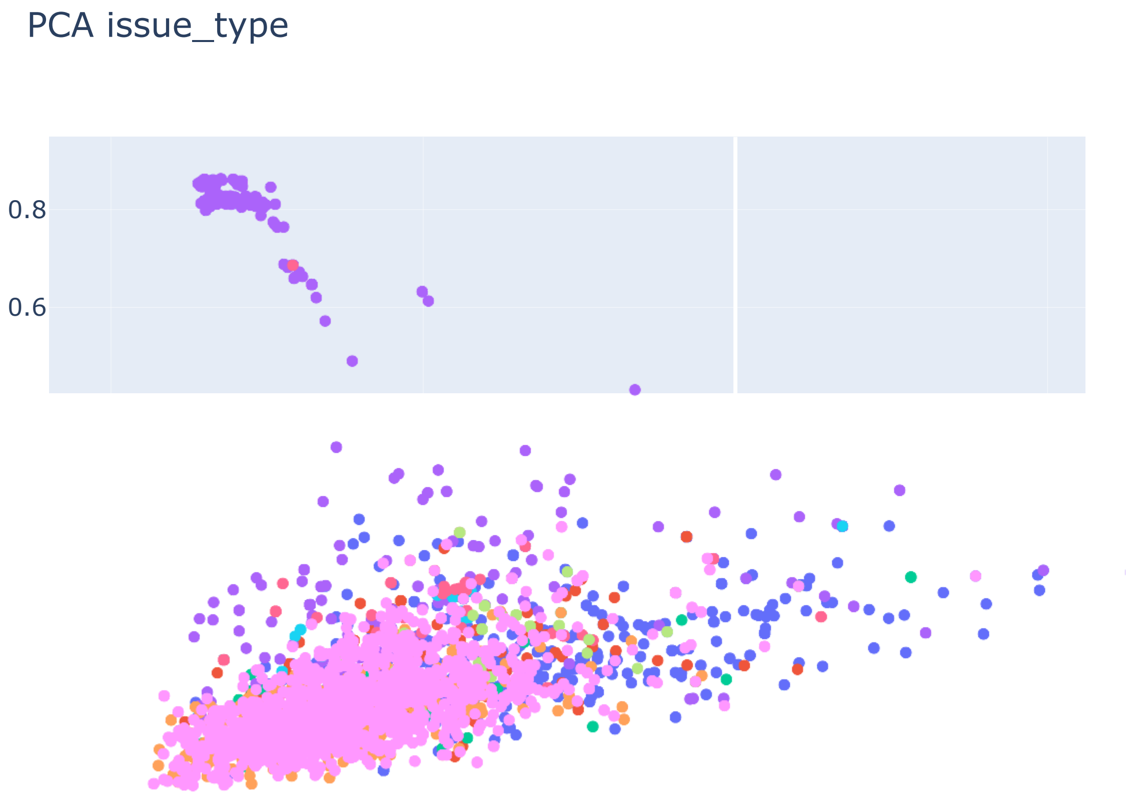
```python
#data = masterData.query(" name == 'Vial - liquid' ").copy()
data = masterData
valueCount = data["issue_type"].value_counts(normalize=True).to_frame().cumsum()*100
data = data[data["issue_type"].isin(valueCount.index.tolist()[0:9])]
data = data[["notes","issue_type","len"]]

data['notes']=data['notes'].apply(lambda cw : clean_notes(cw))
data = data.query(" len > 500 ")
```

```
data["notes"] = data["notes"].apply(lambda x: ' '.join(simple_preprocess(x, min_len=4,
data["notes"] = data["notes"].apply(lambda x: remove_stopwords(''.join(x)))
```

In [22]:

```
data["pca"] = (data["notes"].pipe(representation.tfidf, max_features=100).pipe(represe
hero.scatterplot(data, col="pca", color="issue_type", title="PCA issue_type")
```



In [23]:

```
data["labeled_notes"] = data["issue_type"].apply(lambda x: '__label__' + x + " " ) + c

x_train,x_test,y_train,y_test = train_test_split(data[["labeled_notes","issue_type"]],
x_train.to_csv("C:/Users/gmodi/Downloads/x_train.csv",index=False,header=False)
x_test.to_csv("C:/Users/gmodi/Downloads/x_test.csv",index=False,header=False)

model = fasttext.train_supervised(input="C:/Users/gmodi/Downloads/x_train.csv", wordNg
model.test("C:/Users/gmodi/Downloads/x_test.csv",k=3)
```

Out[23]:

```
(2341, 0.330485547486829, 0.9914566424604869)
```

In [24]:

```
# predict the data
x_test["predicted"] = x_test["labeled_notes"].apply(lambda x: model.predict(x)[0][0]).

#Create the confusion matrix
confusion_matrix(x_test["issue_type"], x_test["predicted"])

print(classification_report(x_test["issue_type"], x_test["predicted"]))
print(confusion_matrix(x_test["issue_type"], x_test["predicted"]))
multiclass_roc_auc_score(x_test["issue_type"], x_test["predicted"])
```

```
                                      precision    recall  f1-score   support

         activation_difficulty_resolved       0.98      0.90      0.94       109
     autoinjector_activation_difficulty       0.95      0.95      0.95       921
           carton_cosmetic_minor_damage       0.99      0.95      0.97       380
                      customer_feedback       0.90      0.95      0.92       515
                         drug_injection       0.72      0.84      0.78        62
                         drug_particles       0.67      0.92      0.78        61
software_based_device_result_incorrect       0.95      0.78      0.86       151
 software_based_device_technical_issue       1.00      0.71      0.83        28
         vial_stopper_damaged_defective       0.99      0.98      0.99       114

                               accuracy                           0.93      2341
                              macro avg       0.91      0.89      0.89      2341
                           weighted avg       0.94      0.93      0.93      2341

[[ 98   9   0   2   0   0   0   0   0]
 [  2 875   1  18   5  20   0   0   0]
 [  0  12 361   0   0   7   0   0   0]
 [  0   8   0 489  12   0   6   0   0]
 [  0   8   0   2  52   0   0   0   0]
 [  0   1   1   0   2  56   0   0   1]
 [  0   3   0  30   0   0 118   0   0]
 [  0   5   0   3   0   0   0  20   0]
 [  0   1   0   0   1   0   0   0 112]]
```

Out[24]:

```
0.9385917238867003
```

In [25]:

```
x_test["prediction"] = x_test["labeled_notes"].apply(lambda x: model.predict(x,3)).ast
#x_test["prediction"] = x_test["prediction"].astype(str)
#x_test["prediction"] = x_test["prediction"].str.replace('__label__','')

x_test["predicted"] = ""
for i in range(len(x_test)):
    if x_test.issue_type.iloc[i] in x_test.prediction.iloc[i]: x_test.predicted.iloc[i
    else: x_test.predicted.iloc[i] = 0

#x_test.to_csv("C:/Users/gmodi/Downloads/x_test_results.csv")
x_test["predicted"].value_counts(normalize=True)*100
```

Out[25]:

```
1    98.077745
0     1.922255
Name: predicted, dtype: float64
```

In [26]:

```
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_Vial_liquid.bin'
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_Vial_liquid.ftz'
```

Vial - lyophilized

In [27]:

```
#data = masterData.query(" name == 'Vial - Lyophilized' ").copy()
data = masterData
valueCount = data["issue_type"].value_counts(normalize=True).to_frame().cumsum()*100
data = data[data["issue_type"].isin(valueCount.index.tolist()[0:14])]
data = data[["notes","issue_type","len"]]
```

```
data['notes']=data['notes'].apply(lambda cw : clean_notes(cw))
data = data.query(" len > 200 ")

data["notes"] = data["notes"].apply(lambda x: ' '.join(simple_preprocess(x, min_len=4,
data["notes"] = data["notes"].apply(lambda x: remove_stopwords(''.join(x)))
```

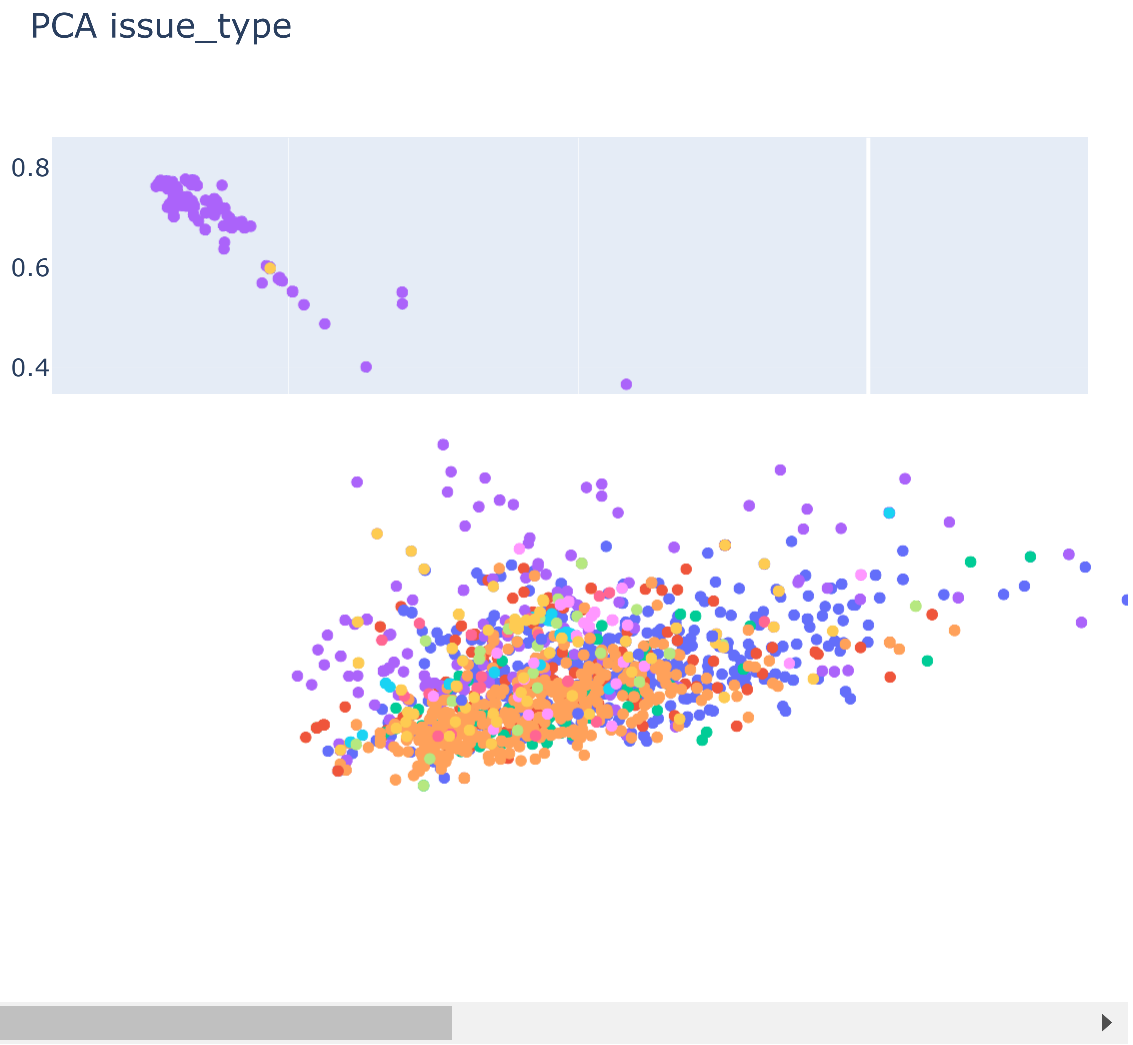
In [28]:
```
valueCount.head(20)
```

Out[28]:

	issue_type
autoinjector_activation_difficulty	33.503568
customer_feedback	54.691285
carton_cosmetic_minor_damage	68.840437
software_based_device_result_incorrect	74.680809
vial_stopper_damaged_defective	78.511098
activation_difficulty_resolved	82.151509
drug_injection	84.718130
drug_particles	86.767498
software_based_device_technical_issue	88.181759
drug_appearance	89.170431
needle_missing	90.080534
vial_plastic_cap_damaged_defective	90.689452
other	91.291822
interface_vial_adapter_leakage_breakage	91.782885
needle_blister_damaged_defective	92.254305
interface_needle	92.692988
drug_fill_volume	93.098933
vial_kit_user_mishandling_difficulty	93.478688
autoinjector_user_mishandling	93.845348
Before_Activation_resolved	94.205461

In [29]:
```
data = data.query(" issue_type not in ['customer_feedback','other','To_be_determined']
```

In [30]:
```
data["pca"] = (data["notes"].pipe(representation.tfidf, max_features=100).pipe(represe
hero.scatterplot(data, col="pca", color="issue_type", title="PCA issue_type")
```



In [31]:
```
data["labeled_notes"] = data["issue_type"].apply(lambda x: '__label__' + x + " " ) + c

x_train,x_test,y_train,y_test = train_test_split(data[["labeled_notes","issue_type"]],
x_train.to_csv("C:/Users/gmodi/Downloads/x_train.csv",index=False,header=False)
x_test.to_csv("C:/Users/gmodi/Downloads/x_test.csv",index=False,header=False)

model = fasttext.train_supervised(input="C:/Users/gmodi/Downloads/x_train.csv", wordNg
model.test("C:/Users/gmodi/Downloads/x_test.csv",k=3)
```

Out[31]:
```
(2962, 0.33040738239927975, 0.9912221471978393)
```

In [32]:
```
# predict the data
x_test["predicted"] = x_test["labeled_notes"].apply(lambda x: model.predict(x)[0][0]).

print(classification_report(x_test["issue_type"], x_test["predicted"]))
print(confusion_matrix(x_test["issue_type"], x_test["predicted"]))
multiclass_roc_auc_score(x_test["issue_type"], x_test["predicted"])
```

```
                                          precision    recall  f1-score   support

          activation_difficulty_resolved       0.99      0.93      0.96       147
      autoinjector_activation_difficulty       0.96      0.96      0.96      1466
            carton_cosmetic_minor_damage       0.96      0.97      0.97       565
                         drug_appearance       0.46      0.60      0.53        43
                          drug_injection       0.64      0.75      0.69       109
                          drug_particles       0.95      0.79      0.86       107
interface_vial_adapter_leakage_breakage       0.68      0.94      0.79        18
                          needle_missing       1.00      1.00      1.00        35
 software_based_device_result_incorrect       0.92      0.91      0.92       223
  software_based_device_technical_issue       0.81      0.67      0.74        52
      vial_plastic_cap_damaged_defective       0.96      0.96      0.96        27
          vial_stopper_damaged_defective       1.00      0.98      0.99       170

                                accuracy                           0.94      2962
                               macro avg       0.86      0.87      0.86      2962
                            weighted avg       0.94      0.94      0.94      2962

[[ 137    9    0    0    1    0    0    0    0    0    0    0]
 [    1 1410   11    6   34    0    2    0    2    0    0    0]
 [    0   17  548    0    0    0    0    0    0    0    0    0]
 [    1    4    2   26    4    3    3    0    0    0    0    0]
 [    0   15    0    4   82    0    3    0    4    1    0    0]
 [    0    1    2   15    3   84    0    0    0    1    1    0]
 [    0    0    1    0    0    0   17    0    0    0    0    0]
 [    0    0    0    0    0    0    0   35    0    0    0    0]
 [    0    3    1    4    5    0    0    0  204    6    0    0]
 [    0    2    2    1    0    0    0    0   12   35    0    0]
 [    0    1    0    0    0    0    0    0    0    0   26    0]
 [    0    1    1    0    0    1    0    0    0    0    0  167]]
```

Out[32]:

```
0.933262441260491
```

In [33]:

```python
x_test["prediction"] = x_test["labeled_notes"].apply(lambda x: model.predict(x,3)).ast
#x_test["prediction"] = x_test["prediction"].astype(str)
#x_test["prediction"] = x_test["prediction"].str.replace('__label__','')

x_test["predicted"] = ""
for i in range(len(x_test)):
    if x_test.issue_type.iloc[i] in x_test.prediction.iloc[i]: x_test.predicted.iloc[i
    else: x_test.predicted.iloc[i] = 0

#x_test.to_csv("C:/Users/gmodi/Downloads/x_test_results.csv")
x_test["predicted"].value_counts(normalize=True)*100
```

Out[33]:

```
1    98.345712
0     1.654288
Name: predicted, dtype: float64
```

In [34]:

```python
model.words
```

```
Out[34]: ['patient',
          '</s>',
          'shipper',
          'aranesp',
          'date',
          'filled',
          'damaged',
          'amgen',
          'issue',
          'product',
          'person',
          'shipment',
          'time',
          'event',
          'complaint',
          'experience',
          'administration',
          'pharmacy',
          'customer',
          'know',
          'wholesaler',
          'receipt',
          'notified',
          'self',
          'open',
          'activated',
          'returned',
          'faulty',
          'available,autoinjector_activation_difficulty',
          'responsible',
          'unfortunately',
          'description',
          'intact',
          'packages',
          'italy',
          'reporter',
          'identified',
          'information',
          'unit',
          'available',
          'injection',
          'received',
          'package',
          'pens',
          'specified',
          'year',
          'reported',
          'expert',
          'dose',
          'vials',
          'vial',
          'email',
          'damage',
          'reports',
          'security',
          'needle',
          'provided',
          'condition',
          'receive',
          'process',
          'delivered',
          'identify',
          'stored',
          'pictures',
          'opened',
          'placed',
          'button',
          'including',
          'till',
          'attached',
          'possession',
          'close',
          'inner',
          'components',
          'knife',
          'sealed',
          'tool',
          'middle',
          'wall',
          'positioned',
          'edge',
          'accordingly',
          'shippers',
          'enbrel',
          'described',
          'batch',
          'number',
          'devices',
          'contact',
          'safety',
          'said',
          'reminders',
          'identified,carton_cosmetic_minor_damage',
          'sureclick',
          'info',
          'replacement',
          'administer',
          'issues',
          'forwarded',
          'adverse',
          'case',
          'following',
          'injections',
          'agent',
          'phone',
          'medication',
          'text',
          'mentioned',
          'took',
          'pushing',
          'manage',
          'observed',
          'states',
          'reminder',
          'germany',
          'intake',
          'consent',
          'functional',
          'called',
          'having',
          'preparation',
          'mail',
          'past',
          'pharmacist',
          'stated',
          'master',
          'place',
          'trying',
          'inject',
          'breda',
          'advised',
          'confirmed',
          'support',
          'federfarna',
          'faulties',
          'intact,carton_cosmetic_minor_damage',
          'thigh',
          'nois',
          'nurse',
          'missed',
          'area',
          'repatha',
          'bergamo',
          'hard',
          'correctly',
          'finding',
          'details',
          'abdominal',
          'like',
          'requested',
          'charge',
          'rubber',
          'today',
          'shakea',
          'shipments',
          'batches',
          'veterinary',
          'administered',
          'surgeon',
          'days',
          'unable',
          'knipperx',
          'comifar',
          'piccirilli',
          'business',
          'able',
          'rossella',
          'cristina',
          'erika',
          'belmont',
          'yeswhen',
          'parsabiv',
          'particles',
          'fusco',
          'chemist',
          'provide',
          'home',
          'healthcare',
          'return',
          'trouble',
          'giulia',
          'sample',
          'brettone',
          'receiving',
          'noticed',
          'fridge',
          'label',
          'syringe',
          'alliance',
          'stuck',
          'knifewere',
          'gets',
          'drug',
          'onnis',
          'mauro',
          'overdose',
          'errors',
          'unexpected',
          'packaging',
          'accidental',
          'exposure',
          'reported,vial_stopper_damaged_defective',
          'feels',
          'january',
          'infectious',
          'therapeutic',
          'benefit',
          'kyprolis',
          'abuse',
          'intentional',
          'transmission',
          'findings',
          'misuse',
          'occupational',
          'signorelli',
          'tried',
          'work',
          'attempt',
          'weeks',
          'giving',
          'advise',
          'notify',
          'circle',
          'device',
          'expected',
          'require',
          'appropriately',
          'inbound',
          'patient,activation_difficulty_resolved',
          'nplate',
          'provided,carton_cosmetic_minor_damage',
          'yeshow',
          'pack',
          'boxes',
          'occasions',
          'sent',
          'hidden',
          'collection,autoinjector_activation_difficulty',
          'report',
          'unifarm',
          'distribuzione',
          'attempted',
          'time,activation_difficulty_resolved',
          'solution',
          'created',
          'totaling',
          'defective',
          'expiry',
          'takes',
          'unico',
          'italia',
          'farm',
          'stefania',
          'logged',
          'note',
          'late',
          'rambelli',
          'alarico',
          'complainant',
          'valentina',
          'vectibix',
          'work,vial_stopper_damaged_defective',
          'yeswhere',
          'address',
          'yesterday',
          'message',
          'click',
          'unpacking',
          'daughter',
          'approximately',
          'informed',
          'pezzullo',
          'successfully',
          'correct',
          'ciro',
          'arcadio',
          'skin',
          'reported,autoinjector_activation_difficulty',
          'prescription',
          'missing',
          'plunger',
          'soon',
          'update',
          'hospital',
          'push',
          'initial',
          'touch',
          'week',
          'prescriber',
          'verify',
          'patients',
          'taken',
          'reporting',
          'awareness',
          'storage,carton_cosmetic_minor_damage',
          'getting',
          'user',
          'greater',
          'point',
          'reference',
          'fulfilled,needle_missing',
          'friday',
          'road',
          'reaching',
          'carer',
          'disposed',
          'working',
          'date,autoinjector_activation_difficulty',
          'happened',
          'retrieve',
          'technician',
          'replied',
          'delay',
          'administering',
          'awaiting',
          'planned',
          'xgeva',
          'draw',
          'collection',
          'world',
          'check',
          'sure',
          'units',
          'phoned',
          'required,autoinjector_activation_difficulty',
          'particle',
          'aware',
          'associated',
          'fawkner',
          'functioning',
          'dispensing',
          'bonwick',
          'obtained,autoinjector_activation_difficulty',
          'arranged',
          'contacted',
          'germany,carton_cosmetic_minor_damage',
          'unknown',
          'follow',
          'portal',
          'completed',
          'different',
          'according',
          'generated',
          'indicated',
          'window',
          'came',
          'white',
          'otezla',
          'disclaimer',
          'visit',
          'receiptwhere',
          'method',
          'worked',
          'goods',
          'notification',
          'taking',
          'afternoon',
          'says',
          'pressed',
          'filing',
          'driving',
          'doctor',
          'presentation',
          'interaction',
          'spain',
          'complete',
          'dates',
          'communication',
          'quality',
          'circle,vial_stopper_damaged_defective',
          'experienced',
          'collect',
          'come',
          'mentioned,autoinjector_activation_difficulty',
          'visiting',
          'sardegna',
          'enrolment',
          'rosita',
          'diluent',
          'alison',
          'care',
          'pharmacies',
          'left',
          'inside',
          'proietto',
          'fabio',
          'coming',
          'cooled',
          'medical',
          'times,activation_difficulty_resolved',
          'provided,vial_plastic_cap_damaged_defective',
          'schiavoni',
          'think',
          'size',
          'blocked,autoinjector_activation_difficulty',
          'given',
          'products',
          'normal',
          'july',
          'difficulty',
          'complaints',
          'prefilled',
          'instead',
          'months',
          'yellow',
          'paglialunga',
          'embark',
          'availablewhen',
          'changed',
          'defect',
          'second',
          'transferred',
          'times',
          'capture',
          'start',
          'outbound',
          'ihre',
          'visible',
          'ihren',
          'timing',
          'refill',
          'delivery',
          'reset',
          'faced',
          'marco',
          'ramella',
          'returnedthree',
          'morning',
          'discount',
          'wednesday',
          'intactis',
          'wife',
          'gubelt',
          'stefano',
          'store',
          'help',
          'started',
          'caller',
          'notifcation',
          'stopper',
          'ministration',
          'provided,vial_stopper_damaged_defective',
          'technical',
          'liquid',
          'better',
          'pierced',
          'later',
          'shopping',
          'complaint,carton_cosmetic_minor_damage',
          'june,autoinjector_activation_difficulty',
          'closed',
          'right',
          'farmacia',
          'returning',
          'reconstituted',
          'village',
          'went',
          'supposed',
          'dosage',
          'technique',
          'sept',
          'request',
          'change',
          'inspected',
          'anna',
          'porge',
          'permission',
          'solution,drug_particles',
          'patrizia',
          'clear',
          'result',
          'municipale',
          'years',
          'password',
          'wrong',
          'aest',
          'stopped',
          'lumykras',
          'refer',
          'injector',
          'account',
          'xxxxwhere',
          'amgevita',
          'workload,autoinjector_activation_difficulty',
          'life',
          'husband',
          'reason',
          'antonella',
          'wanted',
          'preferred',
          'corvino',
          'apply',
          'need',
          'manuela',
          'unusual',
          'pharmaceutical',
          'falerna',
          'assistant',
          'bergantin',
          'select',
          'going',
          'provided,drug_particles',
          'transfer',
          'little',
          'month',
          'birth',
          'cold',
          'error',
          'gironella',
          'outcomes',
          'packs',
          'automated',
          'disability',
          'tuesday',
          'roberta',
          'email,carton_cosmetic_minor_damage',
          'hospitalization',
          'permanent',
          'intervention',
          'complications',
          'threatening',
          'availablewhere',
          'death',
          'heard',
          'pushed',
          'expiration',
          'look',
          'lecce',
          'asked',
          'verified',
          'filamentous',
          'demurtas',
          'injecting',
          'type',
          'appear',
          'carton',
          'discussion',
          'sales',
          'scheduled',
          'fluid',
          'proteinaceous',
          'questions',
          'usual',
          'representant',
          'immediately',
          'marche',
          'send',
          'local',
          'center',
          'cidq',
          'single',
          'removed',
          'requestedwhere',
          'thought',
          'nohow',
          'pain',
          'thursday',
          'sign',
          'sync',
          'allali',
          'bleed',
          'foam',
          'long',
          'remember',
          'biase',
          'dass',
          'read',
          'answer',
          'particulates',
          'want',
          'longer',
          'requesting',
          'review',
          'reported,drug_injection',
          'updated',
          'temperature',
          'veneto',
          'mention',
          'spoke',
          'felt',
          'response',
          'physician',
          'cases',
          'color',
          'receives',
          'starting',
          'determined',
          'half',
          'arrived',
          'autoinjector',
          'stating',
          'depress',
          'sofia',
          'schedule',
          'april',
          'madrid',
          'availablewere',
          'sequence',
          'events',
          'program',
          'spam',
          'reina',
          'ausschlie',
          'bearbeitung',
          'zugriff',
          'informieren',
          'chten',
          'medicine',
          'exact',
          'unternehmen',
          'ihrer',
          'anfrage',
          'reminders,software_based_device_result_incorrect',
          'auswertungen',
          'line',
          'pulse',
          'verwendet',
          'hinaus',
          'gruppe',
          'signed',
          'filled,autoinjector_activation_difficulty',
          'werden',
          'interne',
          'angaben',
          'zusammen',
          'daten',
          'ckfragen',
          'statistische',
          'lich',
          'unpackingwhere',
          'ihnen',
          'napoli',
          'einer',
          'enrolled',
          'datenbank',
          'speichert',
          'gespeichert',
          'mitgeteilten',
          'informationen',
          'namen',
          'kontaktdaten',
          'previous',
          'officer',
          'obtained',
          'thank',
          'fino',
          'tier',
          'asrp,software_based_device_result_incorrect',
          'federfar',
          'share',
          'teams',
          'step',
          'site',
          'messages',
          'cause',
          'pierce',
          'information,autoinjector_activation_difficulty',
          'texts',
          'agglomerates',
          'time,software_based_device_result_incorrect',
          'amorph',
          'problem',
          'injected',
          'good',
          'vendita',
          'unusual,drug_particles',
          'told',
          'checked',
          'twice',
          'medicinali',
          'institution',
          'statwise',
          'remainder',
          'rizzi',
          'silvia',
          'auto',
          'picture',
          'ulssl',
          'file',
          'drawing',
          'story',
          'maintain',
          'difficult',
          'uses',
          'incorrect',
          'blocked',
          'press',
          'obtain',
          'phoenix',
          'occurring',
          'usually',
          'hasn',
          'commitment',
          'saracino',
          'busy',
          'date,carton_cosmetic_minor_damage',
          'trends',
          'mother',
          'failed',
          'needs',
          'blood',
          'farvima',
          'verbal',
          'office',
          'wereused',
          'hear',
          'evaluate',
          'correcher',
          'haben,carton_cosmetic_minor_damage',
          'voicemail',
          'carfilzomib',
          'ingrosso',
          'monday',
          'strange',
          'access',
          'injectors',
          'activation',
          'reply',
          'case,software_based_device_result_incorrect',
          'version',
          'application',
          'data',
          'reconstitution',
          'numbers',
          'additional',
          'requests',
          'elisa',
          'finally',
          'desiree',
          'unsure',
          'montecelo',
          'pontevedra',
          'galicia',
          'xxxwhere',
          'defective,autoinjector_activation_difficulty',
          'accounts',
          'performed',
          'hold',
          'farmaceutici',
          'activated,autoinjector_activation_difficulty',
          'hdpc',
          'excursion',
          'pcms',
          'documents',
          'wellbean',
          'mobile',
          'department',
          'track',
          'seconds',
          'mimpara',
          'login',
          'allowing',
          'december',
          'added',
          'refer,autoinjector_activation_difficulty',
          'android',
          'instructions',
          'idea',
          'fine',
          'couple',
          'limited',
          'provided,software_based_device_result_incorrect',
          'manufacturer',
          'prepared',
          'doses',
          'availablehow',
          'wasn',
          'junk',
          'staff',
          'contacting',
          'away',
          'scad',
          'client',
          'bona',
          'greco',
          'adapter',
          'explained',
          'captured',
          'confused',
          'preparing',
          'needles',
          'receipthow',
          'hands',
          'notice',
          'asti',
          'asrp',
          'hurt',
          'state',
          'march',
          'occurred',
          'turn',
          'jammed',
          'calling',
          'date,software_based_device_result_incorrect',
          'sharps',
          'proceed',
          'patient,software_based_device_result_incorrect',
          'disappear',
          'formed',
          'multiple',
          'incidence',
          'reach',
          'affected',
          'unpacking,carton_cosmetic_minor_damage',
          'investigation',
          'processing',
          'cool',
          'instruction',
          'knifewhere',
          'video',
          'unsuccessful',
          'weekly',
          'required',
          'logistics',
          'training',
          'esercenti',
          'dispense',
          'gauge',
          'unknown,autoinjector_activation_difficulty',
          'moved',
          'september',
          'cooperativa',
          'inspection',
          'gave',
          'looked',
          'beginning',
          'november',
          'obtained,software_based_device_result_incorrect',
          'arizona',
          'eversince',
          'diffenr',
          'syring',
          'prefer,drug_injection',
          'shape',
          'couldn',
          'yesuser',
          'bwas',
          'provider',
          'colorless',
          'syringes',
          'cesare',
          'locandro',
          'brescia',
          'parsabiv,carton_cosmetic_minor_damage',
          'crushed',
          'discarded',
          'screwed',
          'sofarmamorra',
          'feel',
          'quality,software_based_device_result_incorrect',
          'bubbles',
          'partial',
          'injection,software_based_device_result_incorrect',
          'record',
          'refrigeration',
          'extrusion',
          'lotto',
          'today,software_based_device_result_incorrect',
          'symptoms',
          'luisa',
          'inform',
          'mimpara,carton_cosmetic_minor_damage',
          'unicospa',
          'oadded',
          'night',
          'loaded',
          'shaken',
          'leaked',
          'injcetiong',
          'woul',
          'tomorrow',
          'dlike',
          'cahneget',
          'flaky',
          'forth',
          'reviewed',
          'thya',
          'shoul',
          'dnbe',
          'powdery',
          'arounf',
          'sunday',
          'hung',
          'stress',
          'successful',
          'huhg',
          'kept',
          'june',
          'vladimiro',
          'corrected',
          'feis',
          'wouldn',
          'personstefano',
          'comes',
          'continue',
          'internal',
          'previously',
          'floating',
          'turned',
          'recall',
          'followed',
          'nplate,carton_cosmetic_minor_damage',
          'verification',
          'omnicare',
          'lote',
          'settle',
          'lodge',
          'stock',
          'folder',
          'august',
          'prescribed',
          'injection,drug_injection',
          'cell',
          'noweda',
          'appearance',
          'paziente',
          'pantaleo',
          'pell',
          'professional',
          'maione',
          'prior',
          'reibitz',
          'original',
          'probing',
          'attempting',
          'multi',
          'wasted',
          'carmine',
          'maria',
          'street',
          'given,software_based_device_result_incorrect',
          'recently',
          'tablets',
          'autotouch',
          'abdomen',
          'instructed',
          'stop',
          'normally',
          'screen',
          'insert',
          'maybe',
          'galatinamed',
          'specifications,carton_cosmetic_minor_damage',
          'neulasta',
          'thinks',
          'enquiries,autoinjector_activation_difficulty',
          'updates,autoinjector_activation_difficulty',
          'braemar',
          'clarification',
          'release',
          'group',
          'hume,activation_difficulty_resolved',
          'caused',
          'earlier',
          'explains',
          'totally',
          'forgot',
          'container',
          'reset,software_based_device_result_incorrect',
          'setting',
          'tells',
          'goes',
          'gives',
          'appearat',
          'codin',
          'yeshave',
          'workload',
          'particlesare',
          'doriana',
          'lost',
          'bertuccini',
          'difar',
          'escalated',
          'attempts',
          'connect',
          'steps',
          'shot',
          'activate',
          'prescribing',
          'farmalarico',
          'reminder,software_based_device_result_incorrect',
          'early',
          'room',
          'noted',
          'addition',
          'therapy',
          'create',
          'assigned',
          'replaced',
          'pulls',
          'penna',
          'bottle',
          'samples',
          'injects',
          'currently',
          'aedt',
          'arthritis',
          'works',
          'tries',
          'daywhere',
          ...]
```

In [35]:

```
model.labels
```

Out[35]:

```
['__label__autoinjector_activation_difficulty',
 '__label__carton_cosmetic_minor_damage',
 '__label__software_based_device_result_incorrect',
 '__label__vial_stopper_damaged_defective',
 '__label__activation_difficulty_resolved',
 '__label__drug_injection',
 '__label__drug_particles',
 '__label__software_based_device_technical_issue',
 '__label__needle_missing',
 '__label__drug_appearance',
 '__label__vial_plastic_cap_damaged_defective',
 '__label__interface_vial_adapter_leakage_breakage']
```

In [36]:

```
model.wordNgrams
```

Out[36]:

```
4
```

In [37]:
```
model.get_word_vector('drug_particles').shape
```

Out[37]:
```
(100,)
```

In [38]:
```
model.get_nearest_neighbors('overdose')
```

Out[38]:
```
[(0.9925181269645691, 'therapeutic'),
 (0.9921038150787354, 'transmission'),
 (0.9917119741439819, 'intentional'),
 (0.9909931421279907, 'infectious'),
 (0.9909679889678955, 'occupational'),
 (0.990552544593811, 'abuse'),
 (0.9905371069908142, 'misuse'),
 (0.9901981353759766, 'findings'),
 (0.9888617992401123, 'benefit'),
 (0.9865464568138123, 'accidental')]
```

In [39]:
```
model.get_nearest_neighbors('interface_needle')
```

Out[39]:
```
[(0.0, 'aranesp'),
 (0.0, '</s>'),
 (0.0, 'date'),
 (0.0, 'amgen'),
 (0.0, 'filled'),
 (0.0, 'issue'),
 (0.0, 'product'),
 (0.0, 'thursdays'),
 (0.0, 'statement,software_based_device_result_incorrect'),
 (0.0, 'patientt')]
```

In [40]:
```
model.get_nearest_neighbors('syringe')
```

Out[40]:
```
[(0.9349400997161865, 'short'),
 (0.9167957305908203, 'transferred'),
 (0.8911423087120056, 'vial'),
 (0.882398784160614, 'portal,drug_injection'),
 (0.8735360503196716, 'cover'),
 (0.8703376054763794, 'hemoglobin'),
 (0.8688765168190002, 'came,drug_injection'),
 (0.8667225241661072, 'center'),
 (0.8654877543449402, 'spill,drug_injection'),
 (0.8617652654647827, 'spilt')]
```

In [41]:
```
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_Vial_lyophilizec
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_Vial_lyophilizec
```

Software based device

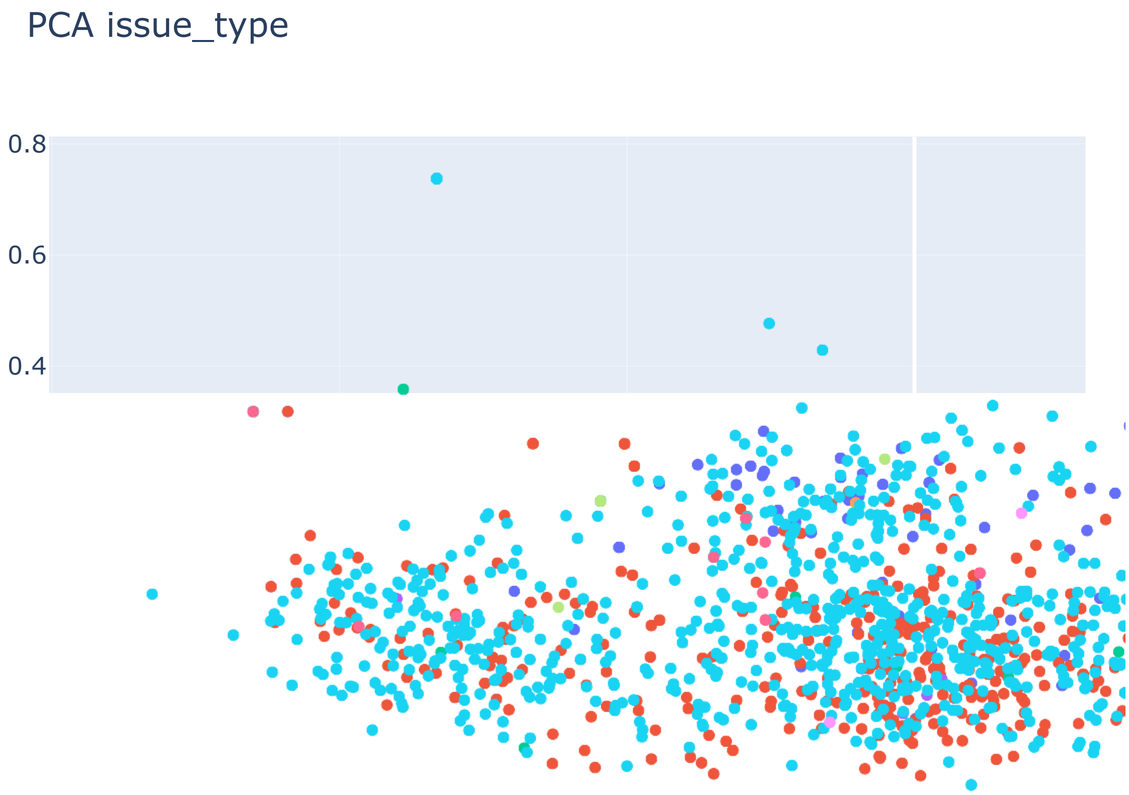
In [42]:
```
data = masterData.query(" name == 'Software based device' ").copy()
valueCount = data["issue_type"].value_counts(normalize=True).to_frame().cumsum()*100
data = data[data["issue_type"].isin(valueCount.index.tolist()[0:9])]
data = data[["notes","issue_type","len"]]

data['notes']=data['notes'].apply(lambda cw : clean_notes(cw))
data = data.query(" len > 500 ")
```

```
data["notes"] = data["notes"].apply(lambda x: ' '.join(simple_preprocess(x, min_len=4,
data["notes"] = data["notes"].apply(lambda x: remove_stopwords(''.join(x)))
```

In [43]:

```
data["pca"] = (data["notes"].pipe(representation.tfidf, max_features=100).pipe(represe
hero.scatterplot(data, col="pca", color="issue_type", title="PCA issue_type")
```



In [44]:

```
data["labeled_notes"] = data["issue_type"].apply(lambda x: '__label__' + x + " " ) + c

x_train,x_test,y_train,y_test = train_test_split(data[["labeled_notes","issue_type"]],
x_train.to_csv("C:/Users/gmodi/Downloads/x_train.csv",index=False,header=False)
x_test.to_csv("C:/Users/gmodi/Downloads/x_test.csv",index=False,header=False)

model = fasttext.train_supervised(input="C:/Users/gmodi/Downloads/x_train.csv", wordNg
model.test("C:/Users/gmodi/Downloads/x_test.csv",k=3)
```

Out[44]:

```
(916, 0.3296943231441048, 0.9890829694323144)
```

In [45]:

```
# predict the data
x_test["predicted"] = x_test["labeled_notes"].apply(lambda x: model.predict(x)[0][0]).

print(classification_report(x_test["issue_type"], x_test["predicted"]))
print(confusion_matrix(x_test["issue_type"], x_test["predicted"]))
multiclass_roc_auc_score(x_test["issue_type"], x_test["predicted"])
```

```
                                       precision    recall  f1-score   support

                 Before_Activation_resolved       1.00      1.00      1.00         1
         autoinjector_activation_difficulty       0.75      0.86      0.80         7
                          customer_feedback       0.88      0.94      0.91       607
                             drug_injection       1.00      0.67      0.80         3
  software_based_device_connectivity_issue       0.00      0.00      0.00         4
               software_based_device_other       0.17      0.67      0.27         3
    software_based_device_result_incorrect       0.84      0.72      0.78       231
     software_based_device_technical_issue       1.00      0.73      0.84        55
       software_user_mishandling_difficulty       1.00      0.20      0.33         5

                                   accuracy                           0.86       916
                                  macro avg       0.74      0.64      0.64       916
                               weighted avg       0.87      0.86      0.86       916

[[  1   0   0   0   0   0   0   0   0]
 [  0   6   0   0   0   0   1   0   0]
 [  0   2 573   0   0   4  28   0   0]
 [  0   0   1   2   0   0   0   0   0]
 [  0   0   0   0   0   4   0   0   0]
 [  0   0   1   0   0   2   0   0   0]
 [  0   0  62   0   0   2 167   0   0]
 [  0   0  15   0   0   0   0  40   0]
 [  0   0   2   0   0   0   2   0   1]]
```

```
C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\sklearn\metrics\_classificat
ion.py:1318: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicte
d samples. Use `zero_division` parameter to control this behavior.

C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\sklearn\metrics\_classificat
ion.py:1318: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicte
d samples. Use `zero_division` parameter to control this behavior.

C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\sklearn\metrics\_classificat
ion.py:1318: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicte
d samples. Use `zero_division` parameter to control this behavior.
```

Out[45]:

```
0.8035630523220642
```

In [46]:

```
x_test["prediction"] = x_test["labeled_notes"].apply(lambda x: model.predict(x,3)).ast
#x_test["prediction"] = x_test["prediction"].astype(str)
#x_test["prediction"] = x_test["prediction"].str.replace('__label__','')

x_test["predicted"] = ""
for i in range(len(x_test)):
    if x_test.issue_type.iloc[i] in x_test.prediction.iloc[i]: x_test.predicted.iloc[i
    else: x_test.predicted.iloc[i] = 0

#x_test.to_csv("C:/Users/gmodi/Downloads/x_test_results.csv")
x_test["predicted"].value_counts(normalize=True)*100
```

Out[46]:

```
1     97.707424
0      2.292576
Name: predicted, dtype: float64
```

In [47]:

```
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_SBD.bin")
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_SBD.ftz")
```

Tablet

In [48]:

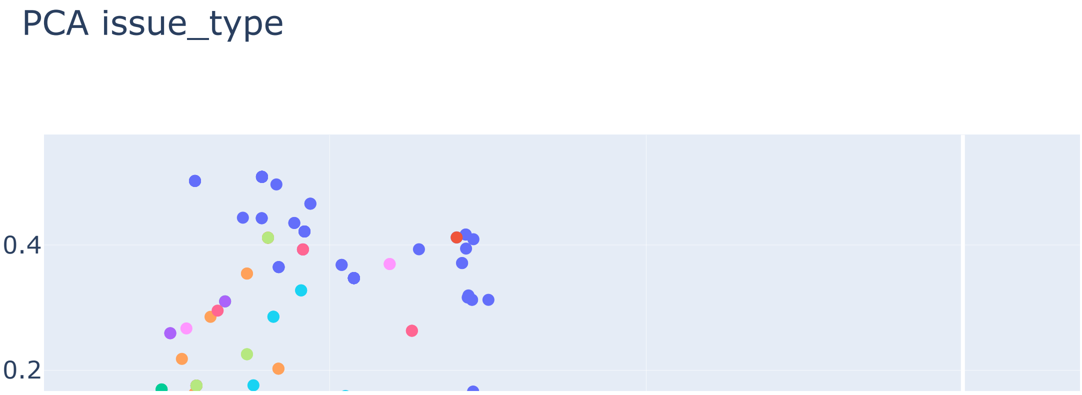
```
data = masterData.query(" name == 'Tablet' ").copy()
valueCount = data["issue_type"].value_counts(normalize=True).to_frame().cumsum()*100
data = data[data["issue_type"].isin(valueCount.index.tolist()[0:9])]
data = data[["notes","issue_type","len"]]

data['notes']=data['notes'].apply(lambda cw : clean_notes(cw))
data = data.query(" len > 500 ")

data["notes"] = data["notes"].apply(lambda x: ' '.join(simple_preprocess(x, min_len=4,
data["notes"] = data["notes"].apply(lambda x: remove_stopwords(''.join(x)))
```

In [49]:

```
data["pca"] = (data["notes"].pipe(representation.tfidf, max_features=100).pipe(represe
hero.scatterplot(data, col="pca", color="issue_type", title="PCA issue_type")
```



```
In [50]: data["labeled_notes"] = data["issue_type"].apply(lambda x: '__label__' + x + " " ) + c

         x_train,x_test,y_train,y_test = train_test_split(data[["labeled_notes","issue_type"]],
         x_train.to_csv("C:/Users/gmodi/Downloads/x_train.csv",index=False,header=False)
         x_test.to_csv("C:/Users/gmodi/Downloads/x_test.csv",index=False,header=False)

         model = fasttext.train_supervised(input="C:/Users/gmodi/Downloads/x_train.csv", wordNg
         print(model.test("C:/Users/gmodi/Downloads/x_test.csv",k=3))

         # predict the data
         x_test["predicted"] = x_test["labeled_notes"].apply(lambda x: model.predict(x)[0][0]).

         print(classification_report(x_test["issue_type"], x_test["predicted"]))
         print(confusion_matrix(x_test["issue_type"], x_test["predicted"]))
         multiclass_roc_auc_score(x_test["issue_type"], x_test["predicted"])
```

```
(163, 0.3231083844580777, 0.9693251533742331)
                             precision    recall  f1-score   support

            To_be_determined       1.00      1.00      1.00         5
     bottle_damaged_defective       0.57      0.80      0.67         5
        bottle_induction_seal       0.25      1.00      0.40         1
        bottle_label_printing       0.00      0.00      0.00         0
              bottle_quantity       1.00      1.00      1.00         2
carton_cosmetic_minor_damage       0.97      1.00      0.99       101
     carton_damaged_defective       0.00      0.00      0.00         5
            customer_feedback       1.00      0.94      0.97        36
              drug_appearance       0.60      0.38      0.46         8

                     accuracy                           0.92       163
                    macro avg       0.60      0.68      0.61       163
                 weighted avg       0.91      0.92      0.91       163

[[  5   0   0   0   0   0   0   0   0]
 [  0   4   1   0   0   0   0   0   0]
 [  0   0   1   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   2   0   0   0   0]
 [  0   0   0   0   0 101   0   0   0]
 [  0   1   1   0   0   1   0   0   2]
 [  0   0   0   0   0   2   0  34   0]
 [  0   2   1   1   0   0   1   0   3]]
```

```
C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\sklearn\metrics\_classificat
ion.py:1318: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true sample
s. Use `zero_division` parameter to control this behavior.

C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\sklearn\metrics\_classificat
ion.py:1318: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true sample
s. Use `zero_division` parameter to control this behavior.

C:\Users\gmodi\Anaconda3\envs\FastText\lib\site-packages\sklearn\metrics\_classificat
ion.py:1318: UndefinedMetricWarning:

Recall and F-score are ill-defined and being set to 0.0 in labels with no true sample
s. Use `zero_division` parameter to control this behavior.
```

Out[50]:
```
0.8758949467280674
```

In [51]:
```
x_test["prediction"] = x_test["labeled_notes"].apply(lambda x: model.predict(x,3)).ast
#x_test["prediction"] = x_test["prediction"].astype(str)
#x_test["prediction"] = x_test["prediction"].str.replace('__label__','')

x_test["predicted"] = ""
for i in range(len(x_test)):
    if x_test.issue_type.iloc[i] in x_test.prediction.iloc[i]: x_test.predicted.iloc[i
    else: x_test.predicted.iloc[i] = 0

#x_test.to_csv("C:/Users/gmodi/Downloads/x_test_results.csv")
x_test["predicted"].value_counts(normalize=True)*100
```

Out[51]:
```
1    98.159509
0     1.840491
Name: predicted, dtype: float64
```

In [52]:
```
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_Tablet.bin")
model.save_model("C:/Users/gmodi/MyProjects/OtherDosageForms/FastText_Tablet.ftz")
```

In [53]:
```
PCM_ISSUES, report_codes = [],[]
```

In [54]:
```
sampleRequest = """ <p><span style="font-size: 10pt;">Wellbean nurse </span><span styl
clean_notes(sampleRequest)
```

Out[54]:
```
'Caller stated they did not have further information to provide except injection site
was on patient s leg The activation button was pressed but the Sureclick pen did not
work There were no click sound no needle penetration and no partial dose received fro
m the complained unit No replacement is required '
```

In [ ]:
```
##### Fast API Main Python File.


from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
import uvicorn
import numpy as np
import pandas as pd
import re
import nltk
from bs4 import BeautifulSoup
import fasttext
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import remove_stopwords

def clean_notes(text):
    soup = BeautifulSoup(text, 'html.parser')
    list1 = [item.get_text() for item in list(soup.children)]
    list2 = [i for i in list1 if len(i) == max([len(i) for i in list1])]
    list3 = [re.sub('[^a-zA-Z:]+', ' ', _) for _ in list2]
    return list3[0]

# Declaring our FastAPI instance
app = FastAPI()

# Defining path operation for root endpoint
@app.get("/")
def main():
    return {
        "message": "Welcome to Amgen AI!"
    }
class request_body(BaseModel):
    AutomationId: str
    DosageForm: str
    Product: str
    ProductID: str
    MasterCase: str
    PCM_Subcase: str
    OccurCountry: str
    PPQ: str
    Notes: str

@app.post("/AMD")
def AMD(data: request_body):
    amd_model = fasttext.load_model("FastText_AMD.ftz")
    issuePredicted = amd_model.predict(clean_notes(data.Notes),k=3)
    PCM_ISSUES, report_codes = [],[]
    for j in range(3):
        report_codes.append({'reported_code': issuePredicted[0][j].replace('__label__'
    PCM_ISSUES.append({'verbatim': list3[0], 'report_codes': report_codes})
    return {
        "AutomationId": data.AutomationId,
        "DosageForm": data.DosageForm,
        "Product": data.Product,
        "ProductID": data.ProductID,
        "MasterCase": data.MasterCase,
        "PCM_Subcase": data.PCM_Subcase,
        "OccurCountry": data.OccurCountry,
        "PPQ": data.PPQ,
        "PCM_ISSUES": PCM_ISSUES
     }
```