Objetivo

- Se busca identificar cual es la percepción en redes sociales de las posiciones politicas en Colombia (Izquiera – Derecha) para ello se realizara un ejercicio de el análisis de sentimientos en la red social Twitter a las cuentas de Gustavo Petro y Alvaro Uribe, lideres políticos de ambas posiciones

Minando data en twitter

Streaming de Twitter

Conectando al API
Mediante tweepy

Obtener tweet en
Archivo Json

ORACLE

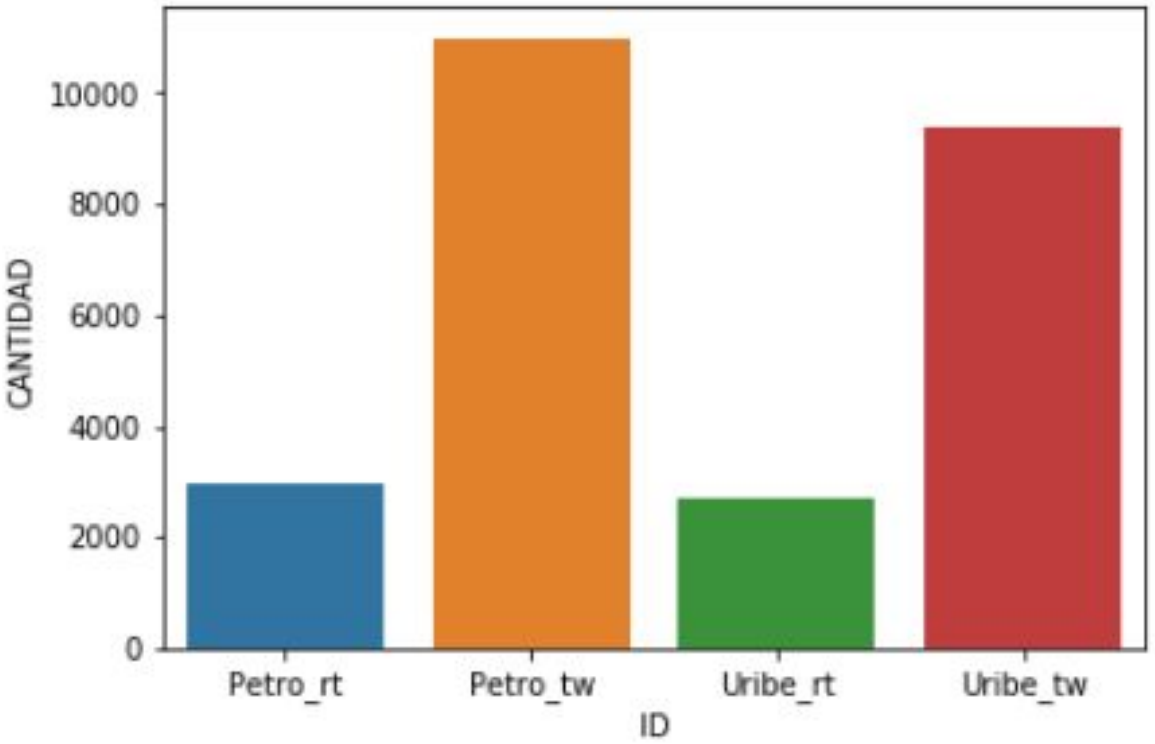
Almacenar data
En db Oracle

Creando CSV con sql

```
create table aux_twitter_lab(
llave_busqueda varchar2(50),
tweet varchar2(500),
usuario varchar2(100),
fecha TIMESTAMP (6) DEFAULT systimestamp,
lugar varchar2(100),
coordenadas varchar2(200),
geo varchar2(100),
json clob)
tablespace ts_table_m;

--*****************************************************************************
--*** Consulta con el fin de identificar la cantidad de tweets por personaje ***
--*****************************************************************************
select llave_busqueda,count(0) conteo from aux_twitter_lab
group by llave_busqueda;

--*************************************************************
--*** Consulta para treaer todo los trinos que no sean reweet ***
--*************************************************************
select
LLAVE_BUSQUEDA,
replace(replace(tweet,chr(10),''),chr(13),'') tweet,
USUARIO,
FECHA,
LUGAR,
COORDENADAS,
GEO
from aux_twitter_lab
where llave_busqueda = 'URIBE'
and tweet NOT like 'RT %';

--*********************************************************
--*** Consulta para treaer todo los trinos que  sean reweet ***
--*********************************************************
select
LLAVE_BUSQUEDA,
replace(replace(tweet,chr(10),''),chr(13),'') tweet,
SUBSTR(tweet,3,INSTR(tweet, ':')-3) USUARIO_RT,
COUNT(0) CONTEO
from aux_twitter_lab
where llave_busqueda = 'PETRO'
and tweet like 'RT %'
AND SUBSTR(tweet,3,INSTR(tweet, ':')-3) IS NOT NULL
GROUP BY LLAVE_BUSQUEDA,
replace(replace(tweet,chr(10),''),chr(13),'') ,
SUBSTR(tweet,3,INSTR(tweet, ':')-3)
ORDER BY 4 DESC;
```

Visualizaciones

Cantidad de tweets y retweets por cada líder político



Wordcloud Petro

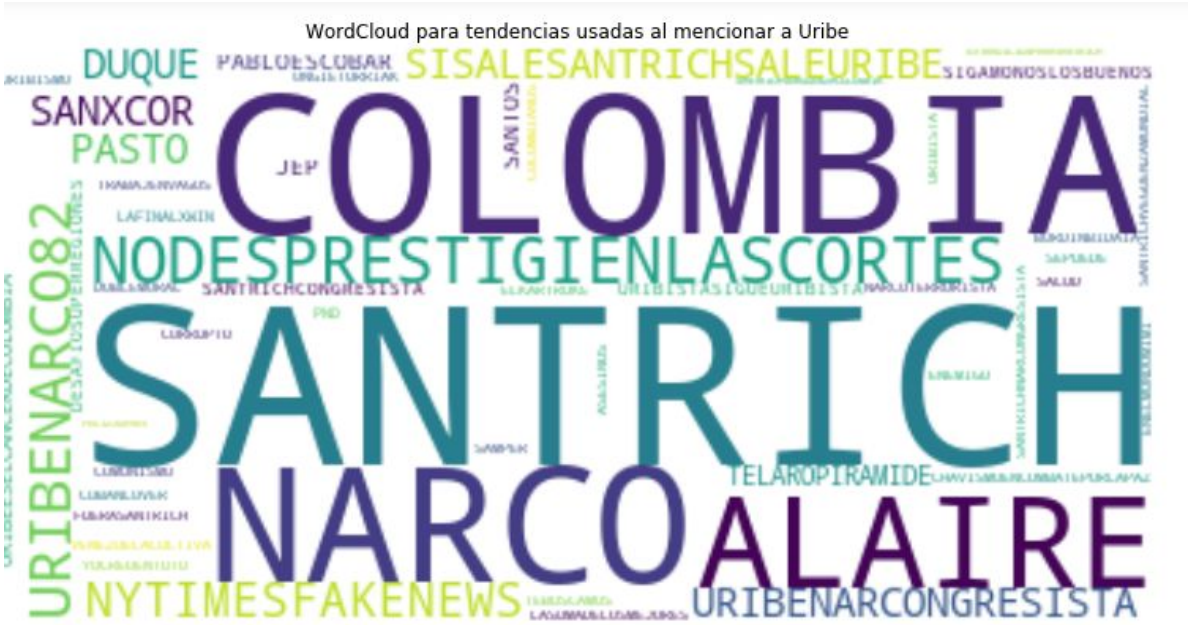
WordCloud para tendencias usadas al mencionar a Petro

WordCloud para usuarios que hablan de Petro

WordCloud para tendencias usadas al mencionar a Petro y tienen retweet

WordCloud para usuarios que hablan de Petro y tienen retweet

Wordcloud Uribe



WordCloud para tendencias usadas al mencionar a Uribe y tienen retweet

WordCloud para usuarios que hablan de Uribe y tienen retweet

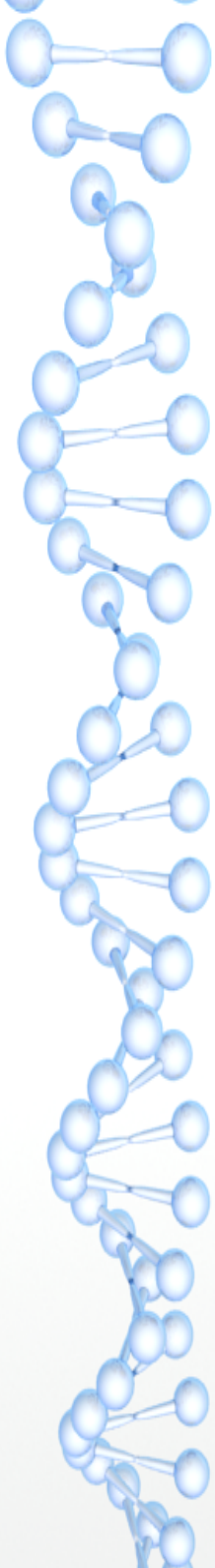
Selección de modelos

Naive_bayes - MultinomialNB

```
from sklearn.naive_bayes import MultinomialNB

text_clf = Pipeline([('vect', count_vectorizer),
                     ('tfidf', TfidfTransformer()),
                     ('clf', MultinomialNB())])

tuned_parameters = {
    'vect__ngram_range': [(1, 1), (1, 2), (2, 2)],
    'tfidf__use_idf': (True, False),
    'tfidf__norm': ('l1', 'l2'),
    'clf__alpha': np.linspace(0.5, 1.5, 6)}

clf = GridSearchCV(text_clf, tuned_parameters, cv=10, n_jobs=5)
clf.fit(Xtrain, ytrain)
```

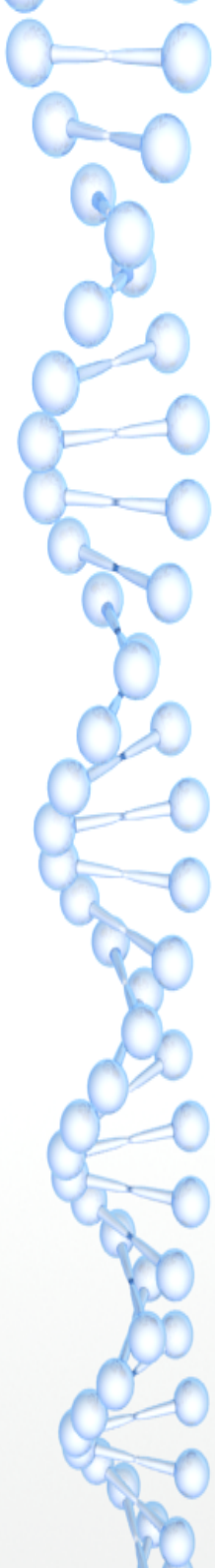
Random Forest

```
from sklearn.ensemble import RandomForestClassifier

n_estimators = [int(x) for x in np.linspace(start = 200, stop = 3000, num = 10)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
bootstrap = [True, False]


text_clf_2 = Pipeline([('vect', count_vectorizer),
                       ('tfidf', TfidfTransformer()),
                       ('clf', RandomForestClassifier())])

tuned_parameters_2 = {
    'vect__ngram_range': [(1, 1), (1, 2), (2, 2)],
    'tfidf__use_idf': (True, False),
    'tfidf__norm': ('l1', 'l2'),
    'clf__n_estimators' : n_estimators,
    'clf__max_features' : max_features,
    'clf__max_depth': max_depth,
    'clf__min_samples_split': min_samples_split,
    'clf__min_samples_leaf': min_samples_leaf,
    'clf__bootstrap': bootstrap}

clf_2 = RandomizedSearchCV(text_clf_2, tuned_parameters_2, cv=10, n_jobs=5, n_iter = 40)
clf_2.fit(Xtrain, ytrain)
```



Suport Vector Machine - SVC

```
from sklearn.svm import SVC

Cs = [0.001, 0.01, 0.1, 1, 10]
gammas = [0.001, 0.01, 0.1, 1]
kernels = ['rbf','sigmoid','linear']


text_clf_3 = Pipeline([('vect', count_vectorizer),
                       ('tfidf', TfidfTransformer()),
                       ('clf', SVC())])



tuned_parameters_3 = {
    'vect__ngram_range': [(1, 1), (1, 2), (2, 2)],
    'tfidf__use_idf': (True, False),
    'tfidf__norm': ('l1', 'l2'),
    'clf__C': Cs,
    'clf__gamma':gammas,
    'clf__kernel':kernels}

clf_3 = GridSearchCV(text_clf_3, tuned_parameters_3, cv=10, n_jobs=5)
clf_3.fit(Xtrain, ytrain)
```



LogisticRegression

```
from sklearn.linear_model import LogisticRegression


text_clf_4 = Pipeline([('vect', count_vectorizer),
                       ('tfidf', TfidfTransformer()),
                       ('clf', LogisticRegression())])



tuned_parameters_4 = {
    'vect__ngram_range': [(1, 1), (1, 2), (2, 2)],
    'tfidf__use_idf': (True, False),
    'tfidf__norm': ('l1', 'l2'),
    'clf__penalty': ['l1', 'l2'],
    'clf__C': [1, 5, 10],
    'clf__max_iter': [20, 50, 100]}

clf_4 = GridSearchCV(text_clf_4, tuned_parameters_4, cv=10, n_jobs=5)
clf_4.fit(Xtrain, ytrain)
```

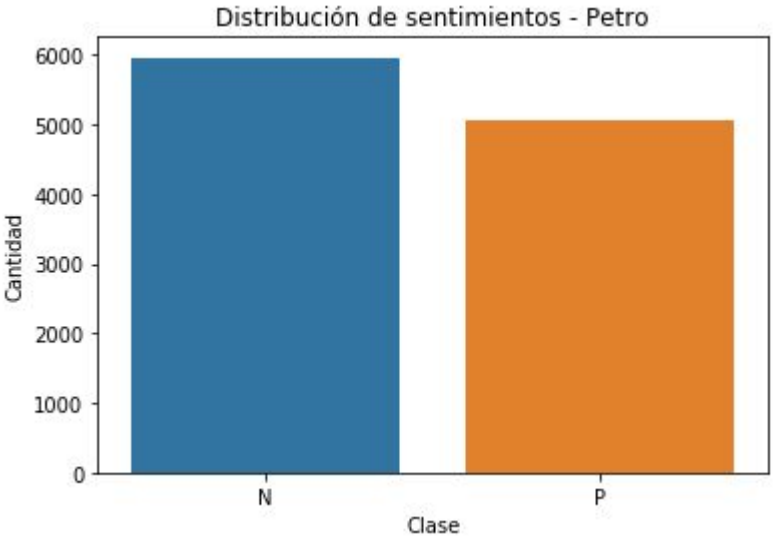
Resultados de los modelos seleccionados

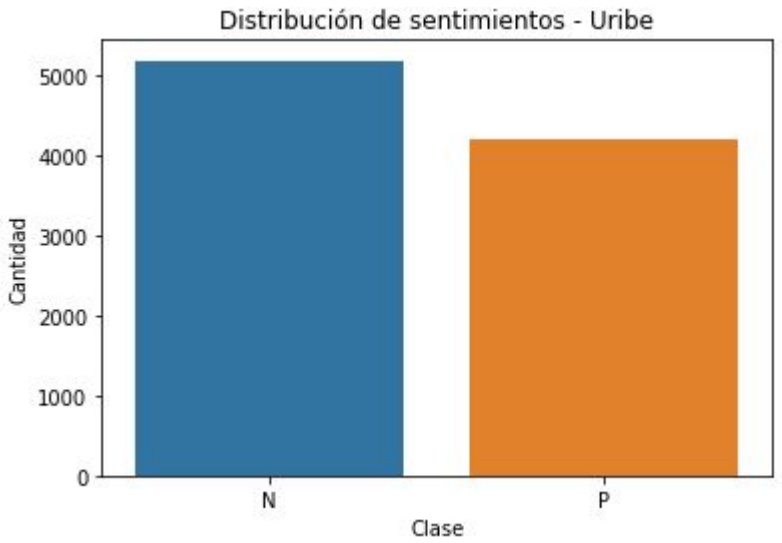
	Modelo	Score	Score_test
2	SVC	0.765730	0.753774
0	MultinomialNB	0.761044	0.752733
3	LogisticRegression	0.764837	0.748048
1	RandomForest	0.732709	0.714732

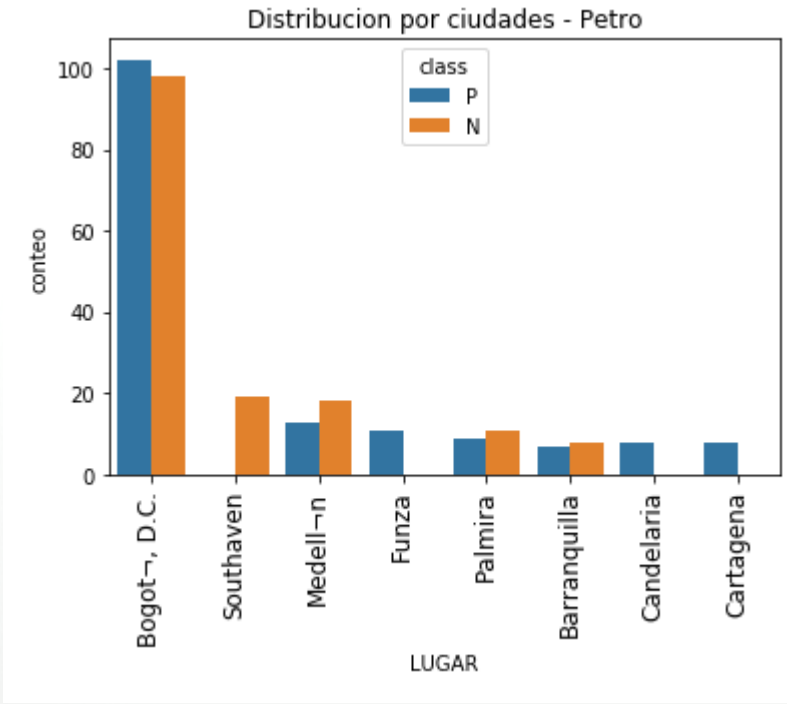
Al parecer el mejor modelo es Svc y seguido por muy poco MultinomialNB de modo que realizaremos el ejercicio con estos dos modelos.

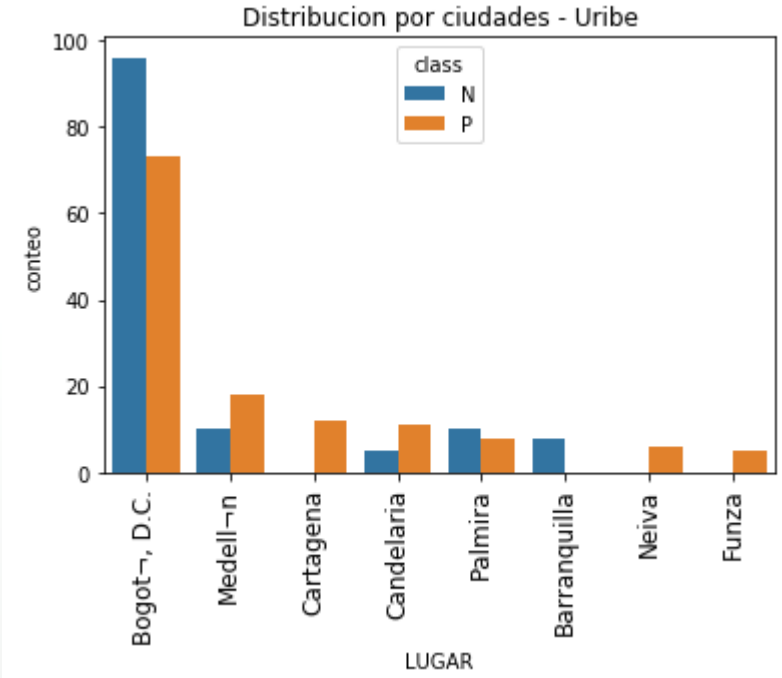
Tambien hace falta resaltar que un score de 75% no es muy bueno

Visualizacion de resultados y clasificacion de tweets









Conclusiones

- 1. El score de los modelos entrenados no supero el 80% por lo que son modelos poco eficientes, esto puede ocurrir porque la data de entrenamiento fue recolectada en España y el dialecto y expresiones son muy diferentes a los usados en Colombia.
- 2. Sin duda los resultados muestran el gran nivel de polarizacion en el pais.
- 3. Segun los diagramas de nube de palabras al parecer cuando un detractor habla mal de un lider politico causa mas retweets que alguien hablando bien de este.
- 4. Bogota es por mucho la ciudad que mas tweetea a estos lideres politicos o al menos la que mas poblacion tiene habilitado el reconocimiento de coordendadas.
- 5. Es necesario crear un set de entrenamiento para el analisis de sentimientos en español y especificamente a nivel Colombia