```
from google.colab import files
uploaded = files.upload()
```

选择文件 未选择任何文件 Upload widget is only available when the cell has been executed in the browser session. Please rerun this cell to enable.

```
Saving arr_big_data.csv to arr_big_data (2).csv
Saving dep_big_data.csv to dep_big_data (1).csv
```

```
import pandas as pd
data = pd.read_csv('arr_big_data.csv')
pd.DataFrame.from_records(data)
del data['Unnamed: 0']
data
```

	presidents_day	easter	memorial_day	independence_day	labor_day	thanksgiving	win
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	
...	...	...	...	...	...	...	
177521	0	0	0	0	0	0	
177522	0	0	0	0	0	0	
177523	0	0	0	0	0	0	
177524	0	0	0	0	0	0	
177525	0	1	0	0	0	0	

177526 rows × 383 columns

Predictions for Arrival Delay

```
import pandas as pd
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import mean_squared_error
import numpy as np

data.dropna(axis=0, subset=['total_arr_delay'], inplace=True)

y = data.total_arr_delay
X = data.drop(['total_arr_delay'], axis=1).select_dtypes(exclude=['object'])


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
```

Mean Absolute Error results

```
data_dmatrix = xgb.DMatrix(data=X,label=y)
```

```
xg_reg = xgb.XGBRegressor(objective ='reg:linear', colsample_bytree = 0.3, learning_rate = 0.1,
                          max_depth = 5, alpha = 10, n_estimators = 10)

xg_reg.fit(X_train,y_train)

preds = xg_reg.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, preds))
print("RMSE: %f" % (rmse))
```

```
RMSE: 23.952019
```

Using k-fold Cross Validation for model tuning

```
params = {"objective":"reg:linear",'colsample_bytree': 0.3,'learning_rate': 0.1,
          'max_depth': 5, 'alpha': 10}

cv_results = xgb.cv(dtrain=data_dmatrix, params=params, nfold=3,
                    num_boost_round=50,early_stopping_rounds=10,metrics="rmse", as_pandas=

cv_results.head()
```

	train-rmse-mean	train-rmse-std	test-rmse-mean	test-rmse-std
0	29.992613	0.076071	29.998277	0.152738
1	28.497772	0.962192	28.505542	0.844652
2	27.752923	1.504529	27.771170	1.478965
3	27.612205	1.511996	27.635091	1.488426
4	26.981518	2.143816	27.014449	2.152146

Better RMSE

```
print((cv_results["test-rmse-mean"]).tail(1))
```

```
49    15.332387
Name: test-rmse-mean, dtype: float64
```

```
xg_reg = xgb.train(params=params, dtrain=data_dmatrix, num_boost_round=10)
```

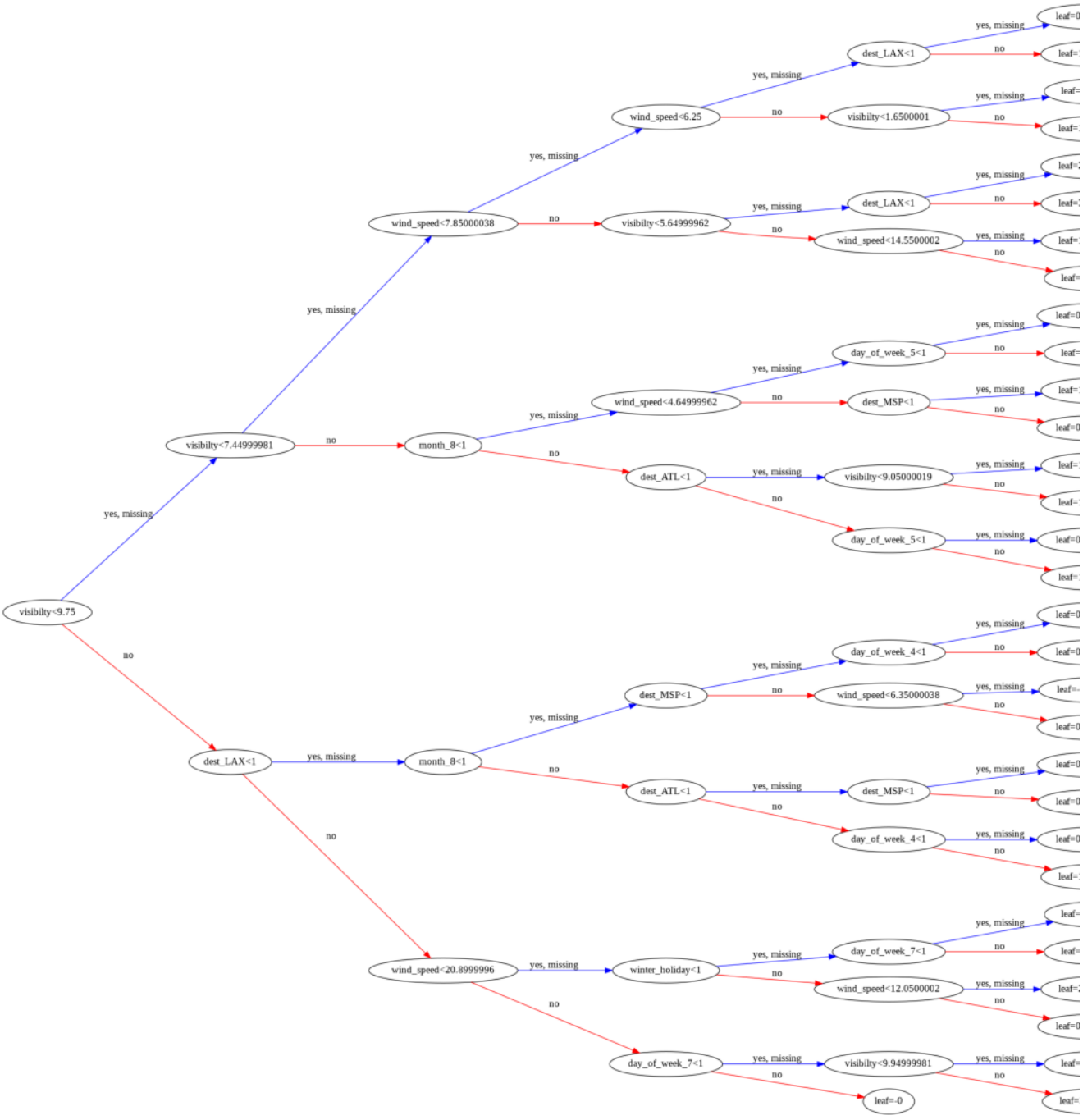
Visualize Boosting Trees and Feature Importance

```
from numpy import loadtxt
from xgboost import XGBClassifier
from xgboost import plot_tree
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = 20,20
plot_tree(xg_reg, num_trees=0, rankdir='LR')
# xgb.plot_tree(xg_reg,num_trees=0)
```

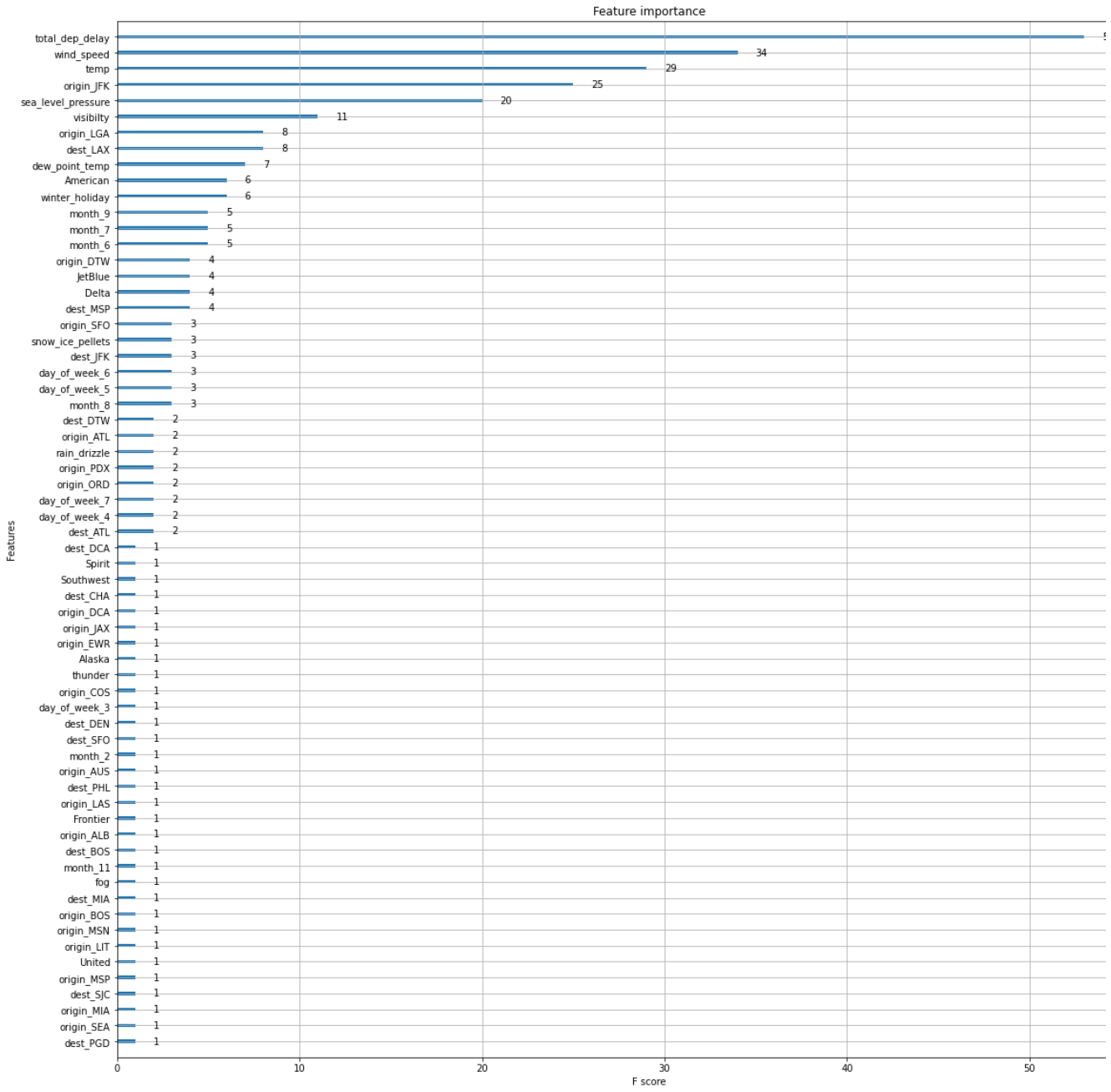
```
#  xgb.plot_tree(xg_reg,  num_trees=0,  rankdir='LR')
#  plt.show()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f896a5644e0>
```



Top Features

```
xgb.plot_importance(xg_reg)
plt.rcParams["figure.figsize"] = 20,20
plt.show()
```



Predictions for Departure Delay

```
import pandas as pd
data = pd.read_csv('dep_big_data.csv')
pd.DataFrame.from_records(data)
```

	distance	presidents_day	easter	memorial_day	independence_day	labor_day	thanksg
0	479	0	0	0	0	0	
1	1222	0	0	0	0	0	
2	2381	0	0	0	0	0	
3	594	0	0	0	0	0	
4	1080	0	0	0	0	0	
...	...	...	...	...	...	...	
177521	594	0	0	0	0	0	
177522	695	0	0	0	0	0	
177523	1182	0	0	0	0	0	
177524	1846	0	0	0	0	0	
177525	546	0	1	0	0	0	

177526 rows × 382 columns

```
import pandas as pd
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import mean_squared_error
import numpy as np

data.dropna(axis=0, subset=['total_dep_delay'], inplace=True)

y = data.total_dep_delay
X = data.drop(['total_dep_delay'], axis=1).select_dtypes(exclude=['object'])


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
```

Mean Absolute Error results

```
data_dmatrix = xgb.DMatrix(data=X,label=y)

xg_reg = xgb.XGBRegressor(objective ='reg:linear', colsample_bytree = 0.3, learning_rate = 0.1,
                          max_depth = 5, alpha = 10, n_estimators = 10)


xg_reg.fit(X_train,y_train)

preds = xg_reg.predict(X_test)


rmse = np.sqrt(mean_squared_error(y_test, preds))
print("RMSE: %f" % (rmse))
```

```
print("RMSE: %f" % (rmse))
```

```
RMSE: 27.676985
```

Using k-fold Cross Validation for model tuning

```
params = {"objective":"reg:linear",'colsample_bytree': 0.3,'learning_rate': 0.1,
                'max_depth': 5, 'alpha': 10}

cv_results = xgb.cv(dtrain=data_dmatrix, params=params, nfold=3,
                    num_boost_round=50,early_stopping_rounds=10,metrics="rmse", as_pandas=

cv_results.head()
```

	train-rmse-mean	train-rmse-std	test-rmse-mean	test-rmse-std
0	33.676422	0.037741	33.679932	0.078407
1	32.415948	0.036744	32.424345	0.083461
2	31.340014	0.037809	31.354115	0.087717
3	30.443820	0.034755	30.463713	0.093509
4	29.686969	0.035026	29.713352	0.096334

Better RMSE

```
print((cv_results["test-rmse-mean"]).tail(1))
```

```
49    25.848924
Name: test-rmse-mean, dtype: float64
```

```
xg_reg = xgb.train(params=params, dtrain=data_dmatrix, num_boost_round=10)
```

Visualize Boosting Trees and Feature Importance

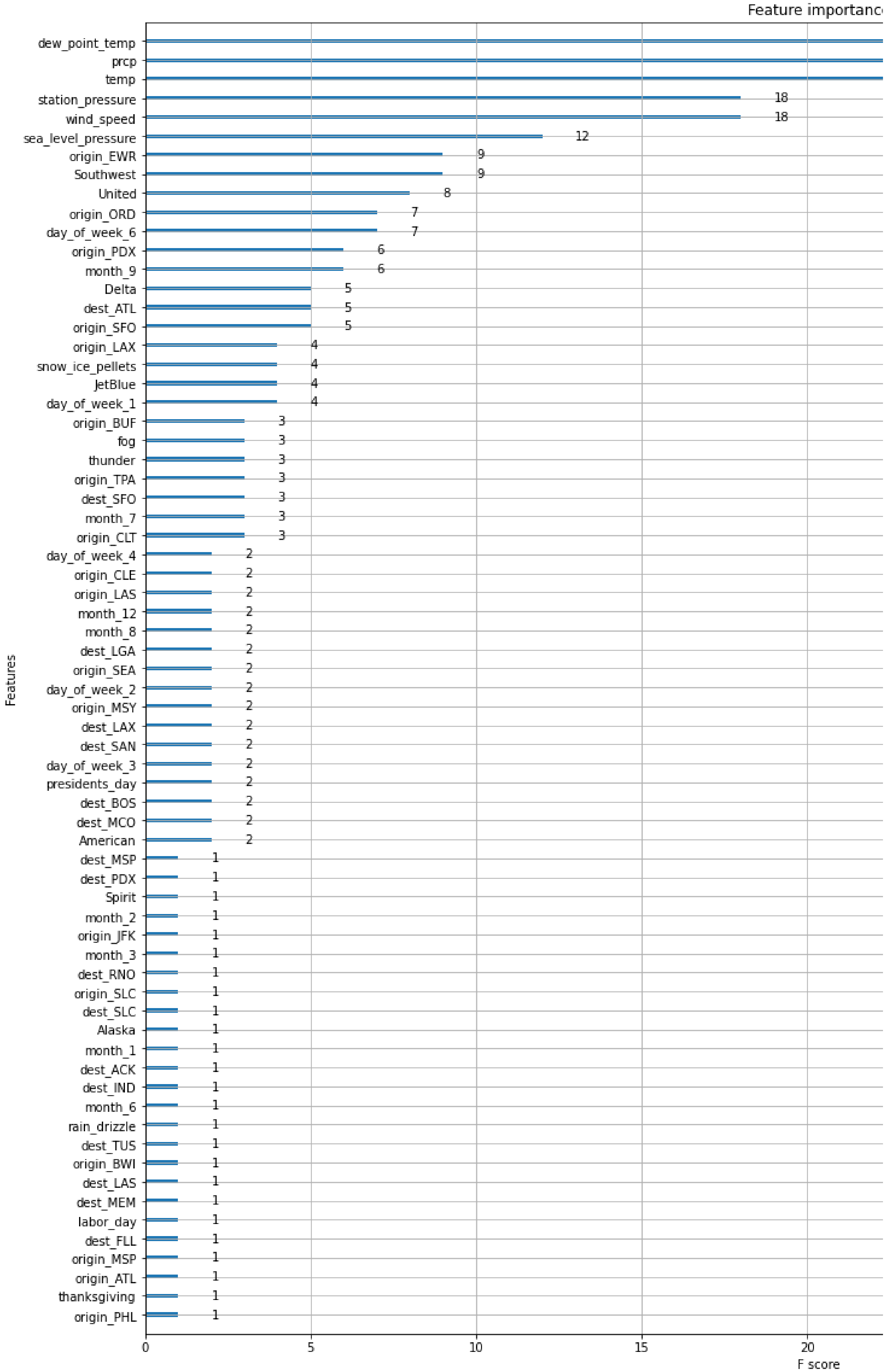
```
from numpy import loadtxt
from xgboost import XGBClassifier
from xgboost import plot_tree
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = 20,20
plot_tree(xg_reg, num_trees=0, rankdir='LR')
#  xgb.plot_tree(xg_reg,num_trees=0)
#  xgb.plot_tree(xg_reg, num_trees=0, rankdir='LR')
#  plt.show()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f895f832ba8>
```

Top Features

```
xgb.plot_importance(xg_reg)
plt.rcParams['figure.figsize'] = [5, 5]
plt.show()
```


Feature importanc
dew_point_temp
prcp
temp
station_pressure 18
wind_speed 18
sea_level_pressure 12
origin_EWR 9
Southwest 9
United 8
origin_ORD 7
day_of_week_6 7
origin_PDX 6
month_9 6
Delta 5
dest_ATL 5
origin_SFO 5
origin_LAX 4
snow_ice_pellets 4
JetBlue 4
day_of_week_1 4
origin_BUF 3
fog 3
thunder 3
origin_TPA 3
dest_SFO 3
month_7 3
origin_CLT 3
day_of_week_4 2
origin_CLE 2
origin_LAS 2
month_12 2
month_8 2
dest_LGA 2
origin_SEA 2
day_of_week_2 2
origin_MSY 2
dest_LAX 2
dest_SAN 2
day_of_week_3 2
presidents_day 2
dest_BOS 2
dest_MCO 2
American 2
dest_MSP 1
dest_PDX 1
Spirit 1
month_2 1
origin_JFK 1
month_3 1
dest_RNO 1
origin_SLC 1
dest_SLC 1
Alaska 1
month_1 1
dest_ACK 1
dest_IND 1
month_6 1
rain_drizzle 1
dest_TUS 1
origin_BWI 1
dest_LAS 1
dest_MEM 1
labor_day 1
dest_FLL 1
origin_MSP 1
origin_ATL 1
thanksgiving 1
origin_PHL 1
Features
0
5
10
15
20
F score