05_Model_Training

April 15, 2022

0.1 Predicting Airline Delays

Notebook: Data Modeling

Team: Jimmy Nguyen, Maha Jayapal, Roberto Cancel

0.2 Packages

[1]:
```
#!pip install --upgrade numpy #ensure numpy and pandas are upgraded to same␣
 ↪versions for easier exploration (avoiding errors)
#!pip install --upgrade pandas #ensure numpy and pandas are upgraded to same␣
 ↪versions for easier exploration (avoiding errors)
!pip install xgboost

import tarfile
import pickle as pkl
import boto3
import sagemaker
from sagemaker import image_uris
from sagemaker.session import Session
from sagemaker.inputs import TrainingInput
import io # for encoding issues with raw data sets
from io import StringIO # converting dataframe to csv and uploading to s3␣
 ↪bucket /tranformed folder

import pandas as pd
import numpy as np
import xgboost
from xgboost import plot_tree
import matplotlib.pyplot as plt
import seaborn as sns
import os
import datetime as dt
import pickle as pkl
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score
```

```
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.py:25:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
Requirement already satisfied: xgboost in /opt/conda/lib/python3.7/site-packages
(1.5.2)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from xgboost) (1.20.3)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages
(from xgboost) (1.4.1)
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommended to
use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is
available.
You should consider upgrading via the '/opt/conda/bin/python -m pip install
--upgrade pip' command.
```

0.3 Set-up

[2]:
```
sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name
```

[3]:
```
print("Default bucket: {}".format(bucket))
```

```
Default bucket: sagemaker-us-east-1-957093009964
```

0.4 Train, Validation, and Test Data

[4]:
```
s3_client = boto3.resource('s3')

# training data
BUCKET = 'ads-508-airline'
KEY = "model_data/df_train.csv"

response = s3_client.Object(BUCKET, KEY)
train = pd.read_csv(response.get()['Body'])
train.head()
```

[4]:

	DEP_DEL15	DAY_OF_WEEK	DEP_TIME_BLK	DISTANCE_GROUP	PILOTS_COPILOTS	\
0	1.0	3	3	4	2840	
1	1.0	3	3	4	2444	
2	1.0	4	4	7	7637	
3	1.0	7	4	5	5175	
4	0.0	5	2	4	7637	

	PASSENGER_HANDLING	PASS_GEN_SVC_ADMIN	MAINTENANCE	PRCP	SNOW	SNWD	\
0	4905	3888	726	0.00	0.0	0.0	
1	23	2273	787	0.29	0.9	1.2	
2	16888	15237	4991	0.00	0.0	1.2	
3	1407	4076	2145	0.04	0.0	0.0	
4	16888	15237	4991	0.00	0.0	0.0	

	TMAX	AWND	WEEK_OF_MONTH
0	50.0	1.57	4
1	41.0	12.30	2
2	27.0	14.99	3
3	67.0	16.11	1
4	57.0	2.01	4

[5]:
```
# validation data
KEY = "model_data/df_val.csv"

response = s3_client.Object(BUCKET, KEY)
valid = pd.read_csv(response.get()['Body'])
valid.head()
```

[5]:

	DEP_DEL15	DAY_OF_WEEK	DEP_TIME_BLK	DISTANCE_GROUP	PILOTS_COPILOTS	\
0	0.0	1	2	4	8989	
1	1.0	4	3	4	8989	
2	0.0	5	4	3	8989	
3	1.0	2	3	8	8989	
4	0.0	7	2	2	8989	

	PASSENGER_HANDLING	PASS_GEN_SVC_ADMIN	MAINTENANCE	PRCP	SNOW	SNWD	\
0	9668	15475	2482	0.0	0.0	0.0	
1	9668	15475	2482	0.0	0.0	0.0	
2	9668	15475	2482	0.0	0.0	0.0	
3	9668	15475	2482	0.0	0.0	0.0	
4	9668	15475	2482	0.0	0.0	0.0	

	TMAX	AWND	WEEK_OF_MONTH
0	42.0	8.05	1
1	60.0	5.82	3
2	61.0	2.68	1
3	55.0	2.91	4

```
4  52.0  9.84              4
```

[6]:
```
# Test data
KEY = "model_data/df_test.csv"

response = s3_client.Object(BUCKET, KEY)
test = pd.read_csv(response.get()['Body'])
test.head()
```

[6]:
```
   DEP_DEL15  DAY_OF_WEEK  DEP_TIME_BLK  DISTANCE_GROUP  PILOTS_COPILOTS  \
0        1.0            1             2               5             8989
1        0.0            4             3               6             7637
2        1.0            4             4               4             8989
3        1.0            6             3               3             8989
4        0.0            2             2               5             8586

   PASSENGER_HANDLING  PASS_GEN_SVC_ADMIN  MAINTENANCE  PRCP  SNOW  SNWD  \
0                9668               15475         2482   0.0   0.0   0.0
1               16888               15237         4991   0.0   0.0   0.0
2                9668               15475         2482   0.0   0.0   0.0
3                9668               15475         2482   0.0   0.0   0.0
4                8586               15502         9677   0.0   0.0   0.0

   TMAX  AWND  WEEK_OF_MONTH
0  53.0  3.36              5
1  68.0  4.70              4
2  80.0  9.40              4
3  41.0  7.61              1
4  50.0  9.17              3
```

0.5 S3 Data Inputs for Modeling

[7]:
```
# Training data
KEY = "model_data/df_train.csv"
s3_input_train = sagemaker.TrainingInput(s3_data='s3://{}/{}'.format(BUCKET,␣
 ↪KEY), content_type='csv')
s3_input_train
```

[7]:
```
<sagemaker.inputs.TrainingInput at 0x7f3917e39910>
```

[8]:
```
# Validation data
KEY = "model_data/df_val.csv"
s3_input_valid = sagemaker.TrainingInput(s3_data='s3://{}/{}'.
 ↪format(BUCKET,KEY), content_type='csv')
```

[9]:
```
# Test data
KEY = "model_data/df_test.csv"
s3_input_test = sagemaker.TrainingInput(s3_data='s3://{}/{}'.format(BUCKET,␣
 ↪KEY), content_type='csv')
```

0.6 Modeling - XGBOOST

[10]:
```
# initialize hyperparameters
hyperparameters = {
        "max_depth":"5", #default 6 - reduced to reduce complexity and␣
 ↪overfitting
        "eta":"0.3", #default
        "gamma":"0", #default
        "min_child_weight":"1", #default
        "subsample":"0.5", #optimized to prevent overfitting
        "lambda":"1", #default
        "objective":"binary:logistic",
        "num_round":"50", "eval_metric":"auc"}

# set an output path where the trained model will be saved
bucket = sagemaker.Session().default_bucket()
prefix = 'baseline_model'
output_path = 's3://{}/{}/{}/output'.format(bucket, prefix, 'xgb-built-in-algo')

# this line automatically looks for the XGBoost image URI and builds an XGBoost␣
 ↪container.
# specify the repo_version depending on your preference.
xgboost_container = sagemaker.image_uris.retrieve("xgboost", region, "1.2-2")

# construct a SageMaker estimator that calls the xgboost-container
estimator = sagemaker.estimator.Estimator(image_uri=xgboost_container,
                                          hyperparameters=hyperparameters,
                                          role=sagemaker.get_execution_role(),
                                          instance_count=1,
                                          instance_type='ml.m5.large',
                                          volume_size=5, # 5 GB
                                          output_path=output_path)

# define the data type and paths to the training and validation datasets
content_type = "libsvm"

# execute the XGBoost training job
estimator.fit({'train': s3_input_train, 'validation': s3_input_valid})
```

```
2022-04-01 17:29:58 Starting - Starting the training
job…ProfilerReport-1648834197: InProgress
…
```

```
2022-04-01 17:31:18 Starting - Preparing the instances for training…
2022-04-01 17:32:58 Downloading - Downloading input data…
2022-04-01 17:33:18 Training - Downloading the training image…
2022-04-01 17:34:59 Training - Training image download completed. Training in
progress.[2022-04-01 17:34:52.616 ip-10-0-126-225.ec2.internal:1 INFO

utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2022-04-01:17:34:52:INFO] Imported framework

sagemaker_xgboost_container.training
[2022-04-01:17:34:52:INFO] Failed to parse hyperparameter eval_metric value

auc to Json.
Returning the value itself
[2022-04-01:17:34:52:INFO] Failed to parse hyperparameter objective value

binary:logistic to Json.
Returning the value itself
[2022-04-01:17:34:52:INFO] No GPUs detected (normal if no gpus

installed)
[2022-04-01:17:34:52:INFO] Running XGBoost Sagemaker in algorithm mode
[2022-04-01:17:34:52:INFO] Determined delimiter of CSV input is ','
[2022-04-01:17:34:52:INFO] Determined delimiter of CSV input is ','
[2022-04-01:17:34:52:INFO] Determined delimiter of CSV input is ','
[2022-04-01:17:34:53:INFO] Determined delimiter of CSV input is ','
[2022-04-01:17:34:53:INFO] Single node training.
[2022-04-01:17:34:53:INFO] Train matrix has 209326 rows and 13 columns
[2022-04-01:17:34:53:INFO] Validation matrix has 11630 rows
[0]#011train-auc:0.64973#011validation-auc:0.64065
[1]#011train-auc:0.65718#011validation-auc:0.65037
[2]#011train-auc:0.66172#011validation-auc:0.65481
[3]#011train-auc:0.66745#011validation-auc:0.66141
[4]#011train-auc:0.67051#011validation-auc:0.66413
[5]#011train-auc:0.67244#011validation-auc:0.66561
[6]#011train-auc:0.67637#011validation-auc:0.66941
[7]#011train-auc:0.68021#011validation-auc:0.67297
[8]#011train-auc:0.68166#011validation-auc:0.67414
[9]#011train-auc:0.68343#011validation-auc:0.67495
[10]#011train-auc:0.68646#011validation-auc:0.67753
[11]#011train-auc:0.68904#011validation-auc:0.68070
[12]#011train-auc:0.68951#011validation-auc:0.68183
[13]#011train-auc:0.69003#011validation-auc:0.68222
[14]#011train-auc:0.69106#011validation-auc:0.68284
[15]#011train-auc:0.69228#011validation-auc:0.68357
[16]#011train-auc:0.69312#011validation-auc:0.68438
[17]#011train-auc:0.69404#011validation-auc:0.68559
[18]#011train-auc:0.69459#011validation-auc:0.68622
[19]#011train-auc:0.69485#011validation-auc:0.68652
[20]#011train-auc:0.69603#011validation-auc:0.68786
```

```
[21]#011train-auc:0.69659#011validation-auc:0.68850
[22]#011train-auc:0.69663#011validation-auc:0.68859
[23]#011train-auc:0.69757#011validation-auc:0.68926
[24]#011train-auc:0.69959#011validation-auc:0.69153
[25]#011train-auc:0.70033#011validation-auc:0.69199
[26]#011train-auc:0.70101#011validation-auc:0.69269
[27]#011train-auc:0.70116#011validation-auc:0.69278
[28]#011train-auc:0.70192#011validation-auc:0.69355
[29]#011train-auc:0.70274#011validation-auc:0.69456
[30]#011train-auc:0.70326#011validation-auc:0.69442
[31]#011train-auc:0.70352#011validation-auc:0.69468
[32]#011train-auc:0.70418#011validation-auc:0.69493
[33]#011train-auc:0.70503#011validation-auc:0.69561
[34]#011train-auc:0.70553#011validation-auc:0.69566
[35]#011train-auc:0.70578#011validation-auc:0.69587
[36]#011train-auc:0.70635#011validation-auc:0.69598
[37]#011train-auc:0.70688#011validation-auc:0.69615
[38]#011train-auc:0.70735#011validation-auc:0.69638
[39]#011train-auc:0.70794#011validation-auc:0.69683
[40]#011train-auc:0.70844#011validation-auc:0.69691
[41]#011train-auc:0.70894#011validation-auc:0.69701
[42]#011train-auc:0.70934#011validation-auc:0.69739

2022-04-01 17:35:29 Uploading - Uploading generated training
model[43]#011train-auc:0.70957#011validation-auc:0.69737
[44]#011train-auc:0.70981#011validation-auc:0.69730
[45]#011train-auc:0.71026#011validation-auc:0.69789
[46]#011train-auc:0.71033#011validation-auc:0.69788
[47]#011train-auc:0.71054#011validation-auc:0.69831
[48]#011train-auc:0.71083#011validation-auc:0.69832
[49]#011train-auc:0.71113#011validation-auc:0.69857

2022-04-01 17:35:59 Completed - Training job completed
ProfilerReport-1648834197: NoIssuesFound
Training seconds: 188
Billable seconds: 188
```

0.7 Evaluation

[11]:
```
# download the model artifact from AWS S3
!aws s3 cp s3://sagemaker-us-east-1-957093009964/baseline_model/
 ↪xgb-built-in-algo/output/sagemaker-xgboost-2022-03-31-00-49-48-555/output/
 ↪model.tar.gz .

#opens the downloaded model artifcat and loads it as 'model' variable
tar = tarfile.open('model.tar.gz')
tar.extractall()
```

```
tar.close()
model = pkl.load(open('xgboost-model', 'rb'))
```

```
download: s3://sagemaker-us-east-1-957093009964/baseline_model/xgb-built-in-
algo/output/sagemaker-xgboost-2022-03-31-00-49-48-555/output/model.tar.gz to
./model.tar.gz
```

0.7.1 Feature Importance:

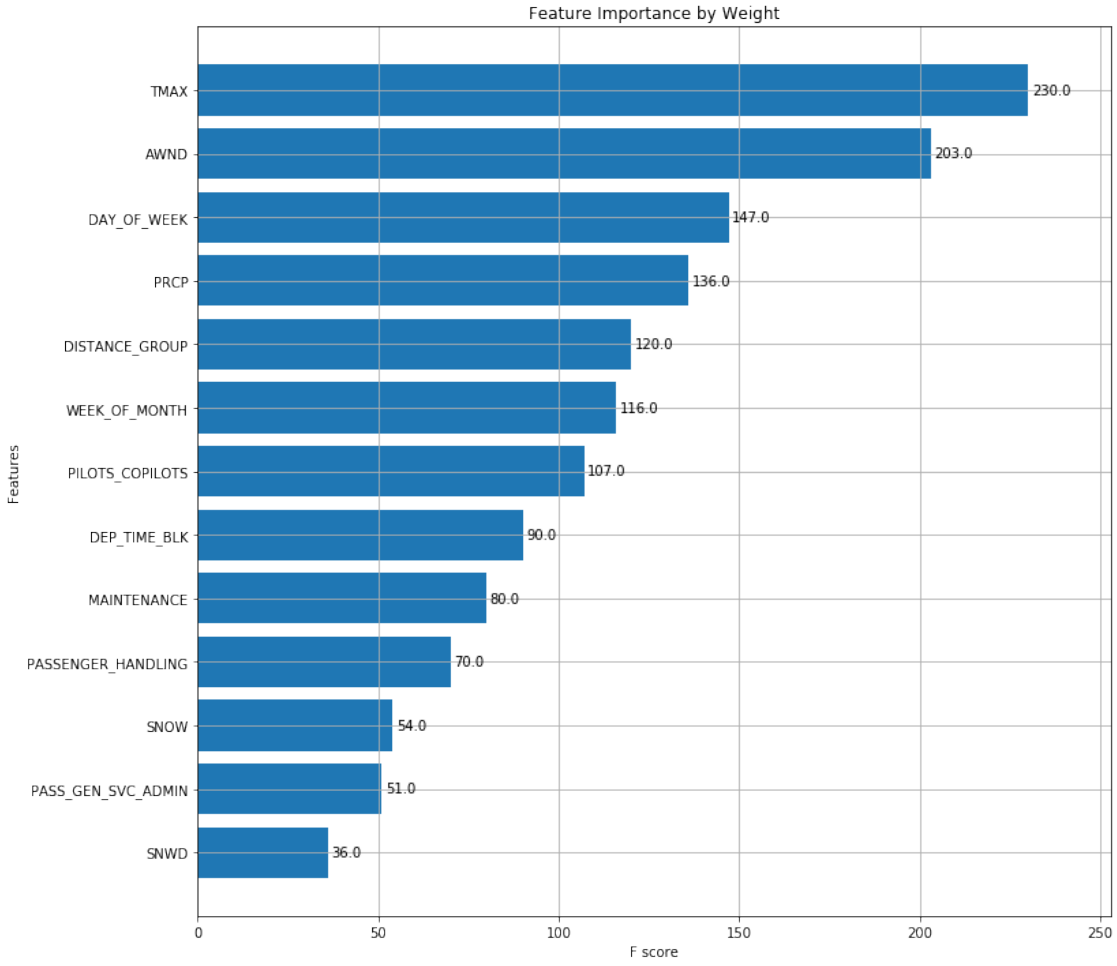
Feature importance is evaluated using weight (the number of times a feature is used to split the data across all trees), cover (the number of times a feature is used to split the data cross all trees weighted by the number of training data points that go through those splits), and gain (the average training loss reduction when using a feature for splitting).

[12]:
```
cols_input = ['DAY_OF_WEEK', 'DEP_TIME_BLK', 'DISTANCE_GROUP',
       'PILOTS_COPILOTS', 'PASSENGER_HANDLING', 'PASS_GEN_SVC_ADMIN',
       'MAINTENANCE', 'PRCP', 'SNOW', 'SNWD', 'TMAX', 'AWND', 'WEEK_OF_MONTH']
cols_target = ['DEP_DEL15']

# Match up with feature names
map_names = dict(zip(list(model.get_fscore().keys()), train[cols_input].
 ↪columns))
model.feature_names = list(map_names.values())

map_names2 = dict(zip(list(model.get_fscore().keys()), train[cols_target].
 ↪columns))
model.target_names = list(map_names2.values())
```

[13]:
```
#plot feature importance with weight
fig, ax = plt.subplots(figsize=(12,12))
xgboost.plot_importance(model, importance_type='weight', max_num_features=30,␣
 ↪height=0.8, ax=ax, show_values = True)
plt.title('Feature Importance by Weight')
plt.show()
```



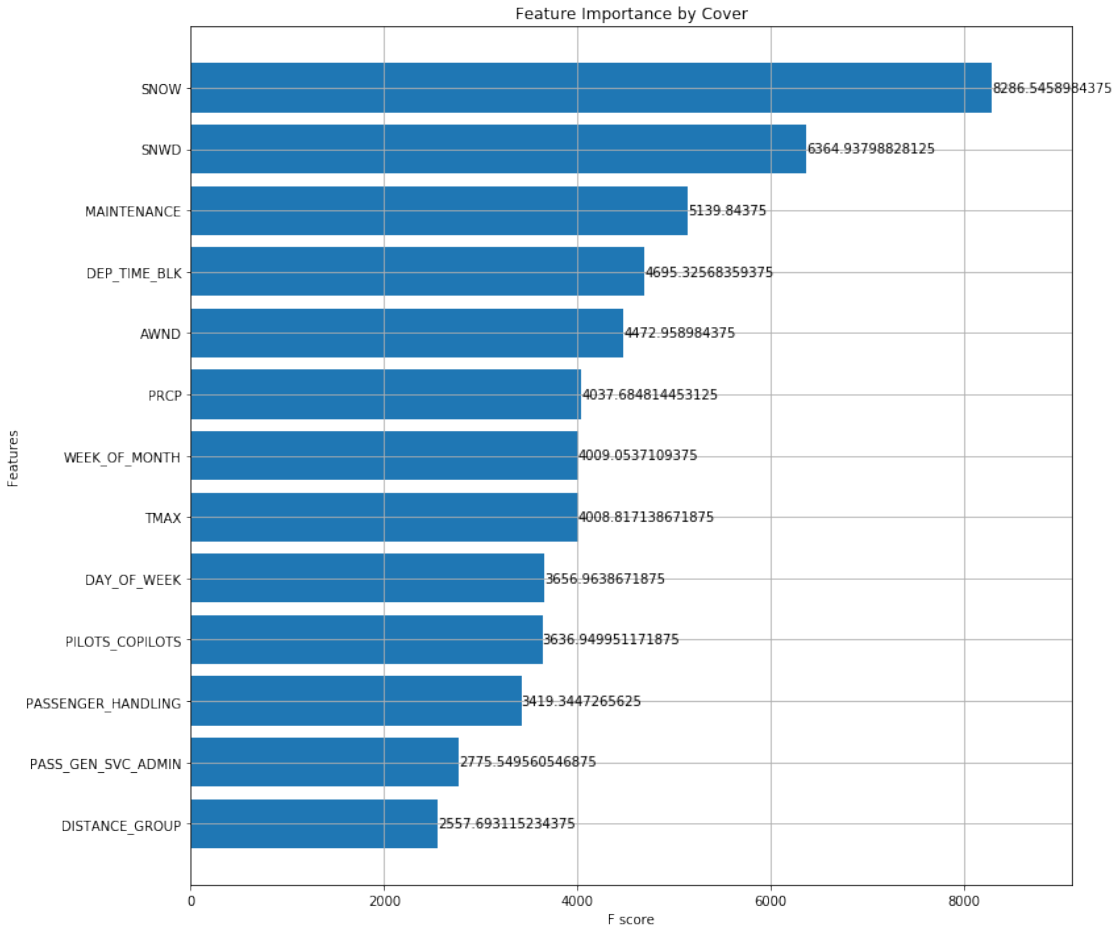
Feature importance by weight ranks features based on the number of times a feature is used to split the data across all trees, we see that the 8 most frequently used features across all trees are: TMAX, AWND, DAY_OF_WEEK, PRCP, DISTANCE_GROUP, WEEK_OF_MONTH, PILOTS_COPILOTS, and DEPT_TIME_BLK.

Summary: Weather conditions, date/time-oriented features, flight length, and staffing levels (pilots/copilots) are the most influential features across all trees at a high-level.

Preliminary Interpretations: While weather conditions aren't features that can be changed by the airline, understanding the interactions of these features with the remaining features would be beneficial. DAY_OF_WEEK, WEEK_OF_MONTH, and DEP_TIME_BLK indicate airport activity and staffing levels are important factors in predicting departure delays.

[14]:
```
#plot feature importance with cover
fig, ax = plt.subplots(figsize=(12,12))
xgboost.plot_importance(model, importance_type='cover', max_num_features=30,␣
↪height=0.8, ax=ax, show_values = True)
plt.title('Feature Importance by Cover')
```

```
plt.show()
```

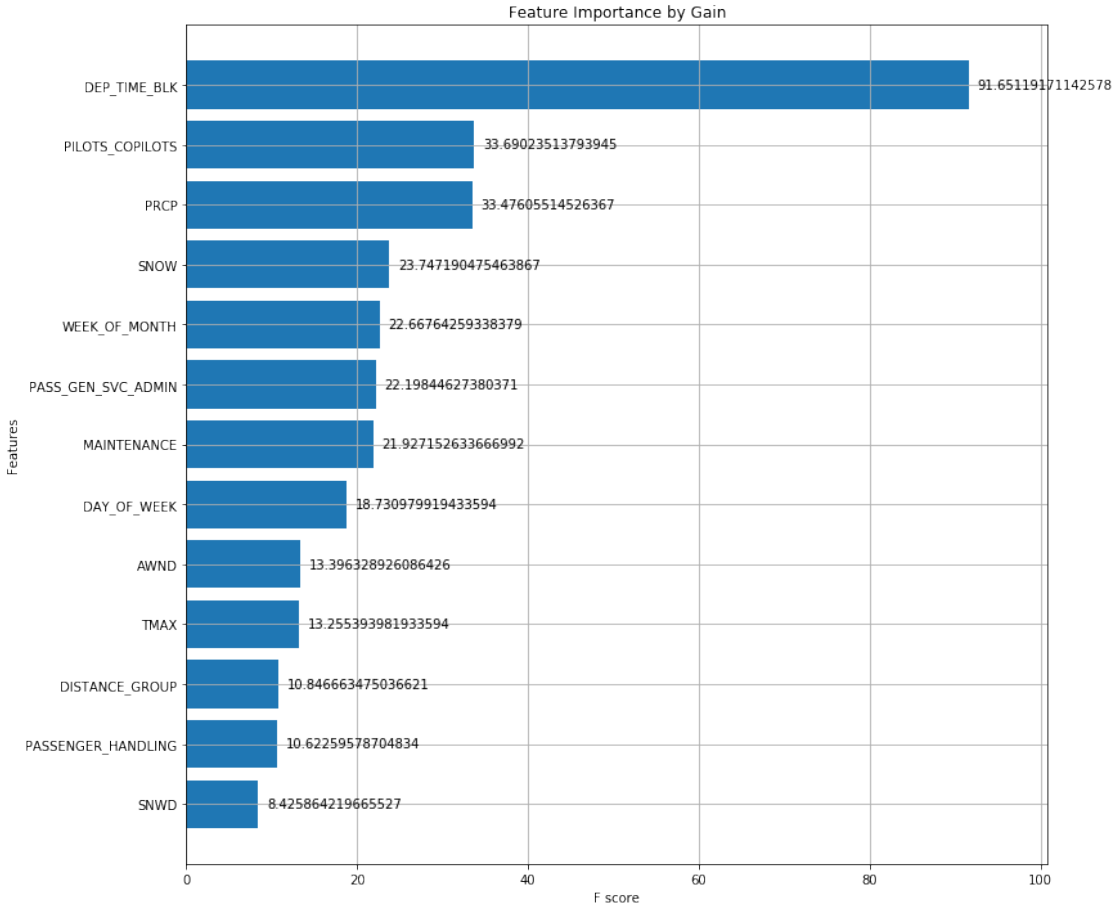


Feature importance by cover ranks features based on the number of times a feature is used to split the data cross all trees weighted by the number of training data points that go through those splits, we see that the 8 most frequently used features are: SNOW, SNWD, MAINTENANCE, DEPT_TIME_BLK, AWND, PRCP, WEEK_OF_MONTH, and TMAX.

Summary: Once again, weather conditions, date/time-oriented features, and staffing levels (maintenance) are the most influential features for splitting trees with respect to the number of training points passed through those splits.

Preliminary Interpretations: We see a similar pattern of features when evaluating by cover.

[15]:
```
#plot feature importance with gain
fig, ax = plt.subplots(figsize=(12,12))
xgboost.plot_importance(model, importance_type='gain', max_num_features=30,␣
↪height=0.8, ax=ax, show_values = True)
plt.title('Feature Importance by Gain')
plt.show()
```



Feature importance by gain ranks features based on the average training loss reduction when using a feature for splitting, we see that the 8 most frequently used features are: DEP_TIME_BLK, PILOTS_COPILOTS, PRCP, SNOW, WEEK_OF_MONTH, PASS_GEN_SVC_ADMIN, MAINTENANCE, and DAY_OF_WEEK.

Summary: Once again, weather conditions, date/time-oriented features, and staffing levels (maintenance) are the most influential features for splitting trees with respect to the number of training points passed through those splits.

0.8 Future Enhancements

[ ]: