

Modeling & Evaluation

- By - Gautam sharma

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import datetime
import shap
import xgboost as xgb
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
```

In [2]:

```
# Show plots in jupyter notebook
%matplotlib inline
```

In [3]:

```
# Set plot style
sns.set(color_codes=True)
```

In [4]:

```
# Set maximum number of columns to be displayed
pd.set_option('display.max_columns', 100)
```

In [5]:

```
# Load JS visualization code to notebook
shap.initjs()
```



In [6]:

```
train_data = pd.read_csv('prepared_data.csv')
```

In [7]:

```
train_data.head(5)
```

Out[7]:

	cons_last_month	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_prod_act	net_
0	10025	0	-41.76	-41.76	1	
1	0	1	25.44	25.44	2	
2	0	0	16.38	16.38	1	
3	0	0	28.60	28.60	1	
4	0	0	30.22	30.22	1	

In [8]:

```
pd.DataFrame({'DataFrame Columns':train_data.columns})
```

Out[8]:

DataFrame Columns

0	cons_last_month
1	has_gas
2	margin_gross_pow_ele
3	margin_net_pow_ele
4	nb_prod_act
5	net_margin
6	num_years_antig
7	pow_max
8	duration
9	price_p1_var
10	price_p2_var
11	price_p3_var
12	price_p1_fix
13	price_p2_fix
14	price_p3_fix
15	churn

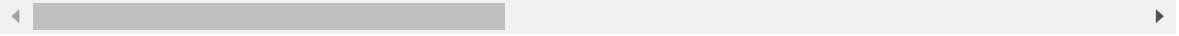
In [9]:

```
y = train_data['churn']
x = train_data.drop(labels= ['churn'], axis = 1)
```

In [10] :

```
x.head(2)
```

Out[10] :

	cons_last_month	has_gas	margin_gross_pow_ele	margin_net_pow_ele	nb_prod_act	net_
0	10025	0	-41.76	-41.76	1	
1	0	1	25.44	25.44	2	
						

In [11] :

```
y.head(5)
```

Out[11] :

```
0      0
1      1
2      0
3      0
4      0
Name: churn, dtype: int64
```

In [12] :

```
x_train, x_test, y_train,y_test = train_test_split(x,y,test_size=0.25, random_state=0)
```

In [13] :

```
x_train.shape
```

Out[13] :

```
(12057, 15)
```

In [14] :

```
x_test.shape
```

Out[14] :

```
(4019, 15)
```

In [15] :

```
model = xgb.XGBClassifier(learning_rate = 0.1, max_depth = 6, n_estimators = 500
, n_jobs = -1)
```

In [16] :

```
model.fit(x_train, y_train)
```

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

[19:13:11] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

Out[16] :

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.1, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints
              ='()', n_estimators=500, n_jobs=-1, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

In [17] :

```
def evaluate(model_, X_test_, y_test_):
    """
    Evaluate the accuracy, precision and recall of a model
    """

    # Get the model predictions
    prediction_test_ = model_.predict(X_test_)

    # Print the evaluation metrics as pandas dataframe
    results = pd.DataFrame({'Accuracy' : [metrics.accuracy_score(y_test_, prediction_test_)], "Precision" : [metrics.precision_score(y_test_, prediction_test_)], "Recall" : [metrics.recall_score(y_test_, prediction_test_)}})

    # For a more detailed report
    #print(metrics.classification_report(y_test_, prediction_test_))
    return results
```

In [18] :

```
evaluate(model, x_test, y_test)
```

Out[18] :

	Accuracy	Precision	Recall
0	0.905947	0.59375	0.097436

ROC-AUC

Reciever operating Charateristics(ROC) curve ia a plot of the true positiverate against the false positive and specify

In [19] :

```
def calculate_roc_auc(model_, x_test_, y_test_):
    """
    Evaluate the roc-auc score
    """

    # Get the model predictions
    # Note that we are using the prediction for the class 1 -> churn
    prediction_test_ = model_.predict_proba(x_test_)[:,1]
    # Compute roc-auc
    fpr, tpr, thresholds = metrics.roc_curve(y_test_, prediction_test_)

    # Print the evaluation metrics as pandas dataframe
    score = pd.DataFrame({"ROC-AUC" : [metrics.auc(fpr, tpr)]})
    return fpr, tpr, score
```

In [20] :

```
def plot_roc_auc(fpr,tpr):
    """
    Plot the Reciever Operating Characteristicfrom a list of the positive rate a
    nd false positive rate.
    """

    # Initialize plot
    f, ax = plt.subplots(figsize=(14,10))

    # Plot ROC
    roc_auc = metrics.auc(fpr, tpr)
    ax.plot(fpr, tpr, lw=2, alpha=0.3,
            label='AUC = %0.2f' % (roc_auc))

    # Plot the randomline.
    plt.plot([0,1],[0,1], linestyle='--', lw=3, color='b',label='Random', alpha=
0.8)

    # Fine tune and show the plot.
    ax.set_xlim([-0.05, 1.05])
    ax.set_ylim([-0.05, 1.05])
    ax.set_xlabel("False Positive Rate (FPR)")
    ax.set_ylabel("True Positive Rate (TPR)")
    ax.set_title("ROC_AUC")
    ax.legend(loc='lower right')
    plt.show
```

In [21] :

```
fpr, tpr, auc_score = calculate_roc_auc(model, x_test, y_test)
```

In [22] :

```
auc_score
```

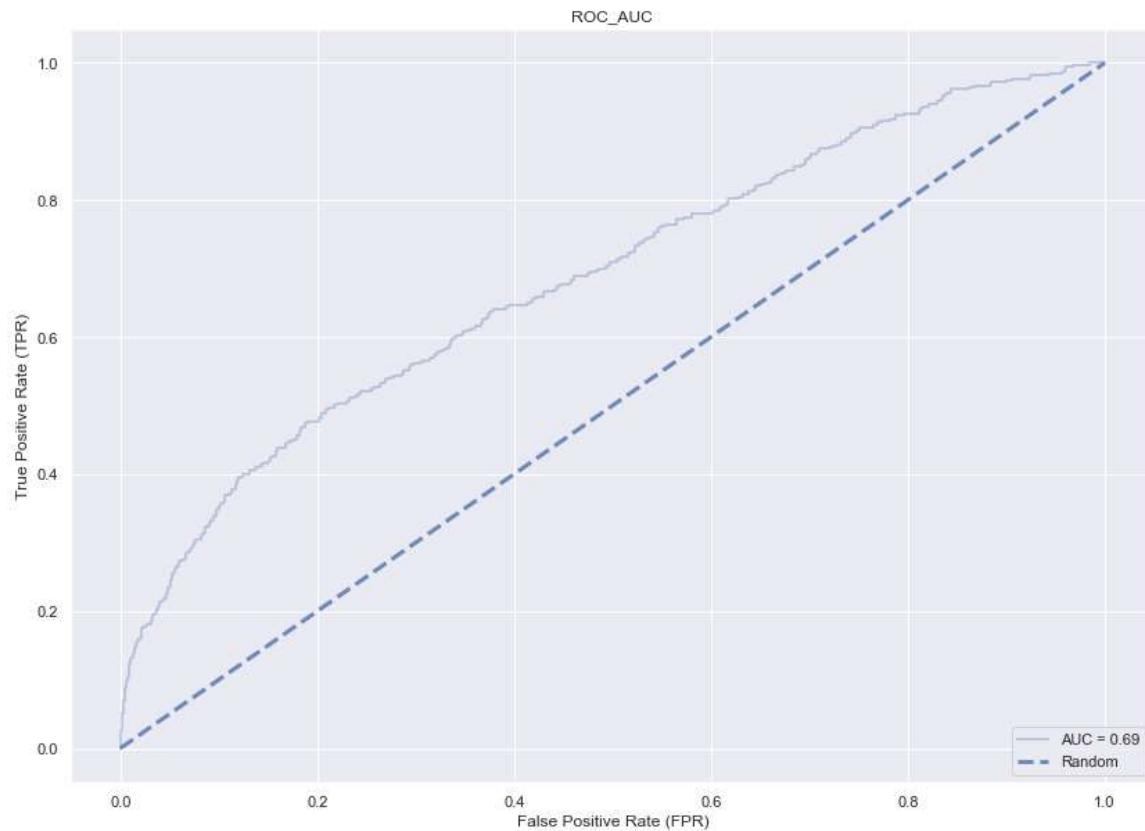
Out [22] :

ROC-AUC

0 0.686587

In [23] :

```
plot_roc_auc(fpr, tpr)
plt.show()
```



Stratified K-fold Validation

ROC_Curve

In [24]:

```
def plot_roc_curve(fprs, tprs):
    """
    Plot the Reciever Operation Charateristics (ROC) from a list of true positive
    rate and false positive rates.
    """

    # Initialize useful lists + the plot axes
    tprs_interp = []
    aucs = []
    mean_fpr = np.linspace(0,1,100)
    f,ax = plt.subplots(figsize=(18,10))

    # Plot ROC for each K-Fold + Compute scores
    for i, (fpr, tpr) in enumerate(zip(fprs, tprs)):
        tprs_interp.append(np.interp(mean_fpr, fpr, tpr))
        tprs_interp[-1][0] = 0.0
        roc_auc = metrics.auc(fpr, tpr)
        aucs.append(roc_auc)
        ax.plot(fpr, tpr, lw=2, alpha=0.3, label='ROC fold %d (AUC = %0.2f)' % (i, roc_auc))

    # Plot the luck line.
    plt.plot([0, 1], [0, 1], linestyle='--', lw=3, color='r',
             label="Random", alpha=.8)

    # Plot the mean ROC.
    mean_tpr = np.mean(tprs_interp, axis=0)
    mean_tpr[-1] = 1.0
    mean_auc = metrics.auc(mean_fpr, mean_tpr)
    std_auc = np.std(aucs)
    ax.plot(mean_fpr, mean_tpr, color='b', label=r"Mean ROC (AUC = %0.2f $\pm$ %0.2f)" % (mean_auc, std_auc), lw=4, alpha=.8)

    # Plot the standard deviation around the mean ROC.
    std_tpr = np.std(tprs_interp, axis=0)
    tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
    tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
    ax.fill_between(mean_fpr, tprs_lower, tprs_upper, color="grey", alpha=.2, label=r"$\pm$ 1 std. dev.")

    # Fine tune and show the plot.
    ax.set_xlim([-0.05, 1.05])
    ax.set_ylim([-0.05, 1.05])
    ax.set_xlabel("False Positive Rate (FPR)")
    ax.set_ylabel("True Positive Rate (TPR)")
    ax.set_title("ROC-AUC")
    ax.legend(loc="lower right")
    plt.show()
    return (f, ax)
```

In [25]:

```
def compute_roc_auc(model_, index):
    y_predict = model_.predict_proba(x.iloc[index])[:,1]
    fpr, tpr, thresholds = metrics.roc_curve(y.iloc[index], y_predict)
    auc_score = metrics.auc(fpr, tpr)
    return fpr, tpr, auc_score
```

In [26]:

```
cv = StratifiedKFold(n_splits = 5, random_state=13, shuffle = True)
```

In [27]:

```
fprs, tprs, scores= [],[],[]
```

In [28]:

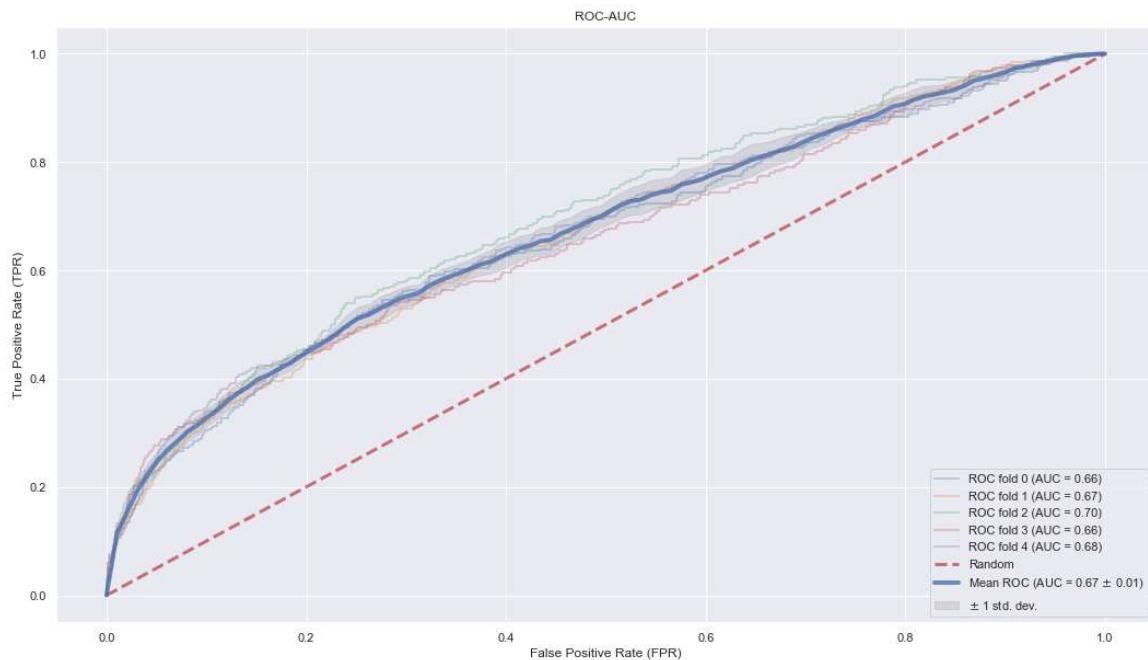
```
for (train, test), i in zip(cv.split(x, y), range(5)):
    model.fit(x.iloc[train], y.iloc[train])
    _, _, auc_score_train = compute_roc_auc(model, train)
    fpr, tpr, auc_score = compute_roc_auc(model, test)
    scores.append((auc_score_train, auc_score))
    fprs.append(fpr)
    tprs.append(tpr)
```

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

[19:13:20] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[19:13:28] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[19:13:35] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[19:13:42] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[19:13:49] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

In [29] :

```
plot_roc_curve(fprs, tprs)
plt.show()
```



Model Finetuning

In [30] :

```
# Random search cross Validation
from sklearn.model_selection import RandomizedSearchCV
```

In [31] :

```
# Create the random grid
params = {
    'min_child_weight': [i for i in np.arange(1,15,1)],
    'gamma': [i for i in np.arange(0,6,0.5)],
    'subsample': [i for i in np.arange(0,1.1,0.1)],
    'colsample_bytree': [i for i in np.arange(0,1.1,0.1)],
    'max_depth': [i for i in np.arange(1,15,1)],
    'scale_pos_weight':[i for i in np.arange(1,15,1)],
    'learning_rate': [i for i in np.arange(0,0.15,0.01)],
    'n_estimators': [i for i in np.arange(0,2000,100)]
}
```

In [32] :

```
# Create model
xgb = xgb.XGBClassifier(objective='binary:logistic', silent=True, nthread=1)
```

In [33] :

```
xgb_random = RandomizedSearchCV(xgb, param_distributions=params,n_iter=1,scoring
='roc_auc',n_jobs=4,
                                cv=5,verbose=3, random_state=1001)
```

In [34] :

```
xgb_random.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
```

```
[Parallel(n_jobs=4)]: Using backend LokyBackend with 4 concurrent workers.
```

```
[Parallel(n_jobs=4)]: Done    2 out of  5 | elapsed:   6.1s remaining: 9.3s
```

```
[Parallel(n_jobs=4)]: Done    5 out of  5 | elapsed:   7.2s finished
```

```
The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
```

```
[19:14:05] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:
```

```
Parameters: { "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but

passed down to XGBoost core. Or some parameters are not used but slip through this

verification. Please open an issue if you find above cases.

```
[19:14:05] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out [34] :

```
RandomizedSearchCV(cv=5,
                     estimator=XGBClassifier(base_score=None, booster=
None,
                                         colsample_bylevel=None,
                                         colsample_bynode=None,
                                         colsample_bytree=None, ga
mma=None,
                                         gpu_id=None, importance_t
ype='gain',
                                         interaction_constraints=N
one,
                                         learning_rate=None,
                                         max_delta_step=None, max_
depth=None,
                                         min_child_weight=None, mi
ssing=nan,
                                         monotone_constraints=None,
                                         n_estimators=100, ...
                                         'min_child_weight': [1, 2,
3, 4, 5, 6,
9, 10, 11,
14],
                                         'n_estimators': [0, 100, 20
0, 300, 400,
00, 800,
1100, 1200,
1500, 1600,
1900],
                                         'scale_pos_weight': [1, 2,
3, 4, 5, 6,
9, 10, 11,
14],
                                         'subsample': [0.0, 0.1, 0.2,
0.300000000000
00004, 0.4,
00000001,
0001, 0.8,
random_state=1001, scoring='roc_auc', verbose=3)
```

In [35]:

```
best_random = xgb_random.best_params_
best_random = {
    'subsample':0.8,
    'scale_pos_weight':1,
    'n_estimators':110,
    'min_child_weight':1,
    'max_depth':12,
    'learning_rate':0.01,
    'gamma':4.0,
    'colsample_bytree':0.60
}
```

In [49]:

```
import xgboost as xgb
```

In [51]:

```
# Create a model with the parameters found
model_random=xgb.XGBClassifier(objective='binary:logistic',silent=True, nthread=1, **best_random)
fprs, tprs, scores = [], [], []
```

In [52]:

```
for (train, test), i in zip(cv.split(x, y), range(5)):
    model_random.fit(x.iloc[train], y.iloc[train])
    _, _, auc_score_train = compute_roc_auc(model_random, train)
    fpr, tpr, auc_score = compute_roc_auc(model_random, test)
    scores.append((auc_score_train, auc_score))
    fprs.append(fpr)
    tprs.append(tpr)
```

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
g: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

```
[19:20:54] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:20:54] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[19:20:59] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:20:59] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[19:21:04] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:21:04] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[19:21:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:21:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the def
```

ault evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

[19:21:14] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:

Parameters: { "silent" } might not be used.

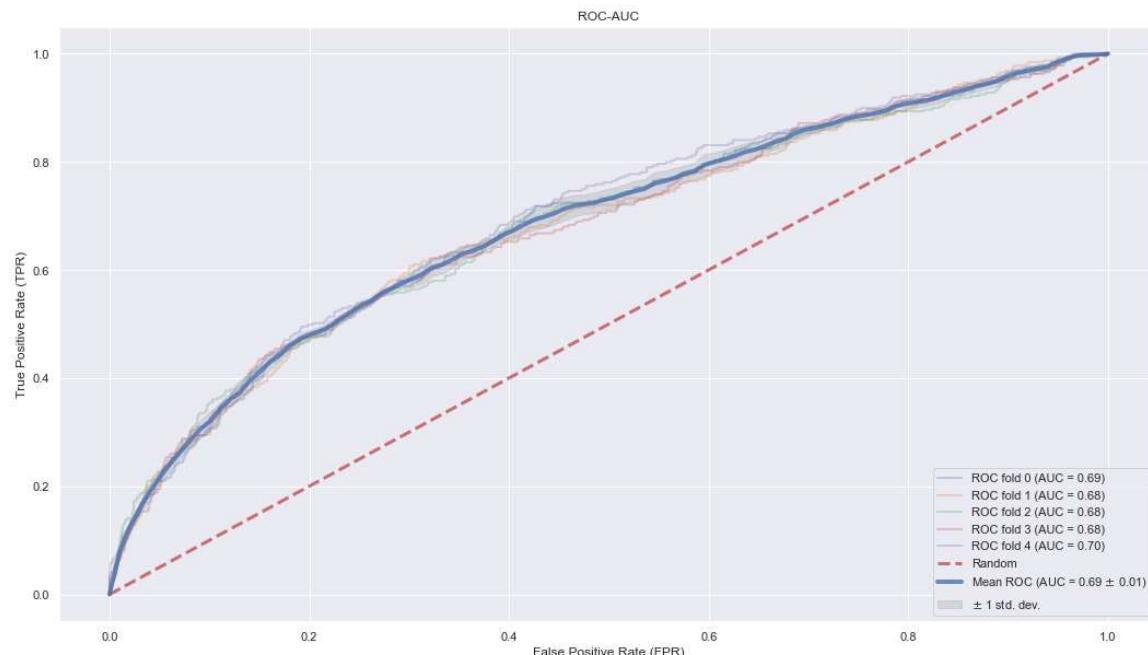
This may not be accurate due to some parameters are only used in language bindings but

passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

[19:21:14] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

In [53]:

```
plot_roc_curve(fprs, tprs)
plt.show()
```



Grid search with cross validation (calculation over weekend, them make smaller)

In [54]:

```
from sklearn.model_selection import GridSearchCV
```

In [55]:

```
# Create the parameter grd based on the results of random search
param_grid = {
    'subsample':[0.7],
    'scale_pos_weight':[1],
    'n_estimators':[1100],
    'min_child_weight':[1],
    'max_depth':[12,13,14],
    'learning_rate':[0.005,0.01],
    'gamma':[4.0],
    'colsample_bytree':[0.6]
}
```

In [56]:

```
xg = xgb.XGBClassifier(objective='binary:logistic',silent=True, nthread=1)
```

In [57]:

```
grid_search = GridSearchCV(estimator = xg, param_grid = param_grid, cv= 5, n_jobs= -1, verbose=2,
                           scoring='roc_auc')
```

In [58]:

```
# Fit the grid search to the data
grid_search.fit(x_train, y_train)
```

Fitting 5 folds for each of 6 candidates, totalling 30 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.

[Parallel(n_jobs=-1)]: Done 30 out of 30 | elapsed: 8.3min finished

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

[19:29:41] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:

Parameters: { "slient" } might not be used.

This may not be accurate due to some parameters are only used in language bindings but

passed down to XGBoost core. Or some parameters are not used but slip through this

verification. Please open an issue if you find above cases.

[19:29:41] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

Out [58] :

```
GridSearchCV(cv=5,
            estimator=XGBClassifier(base_score=None, booster=None,
                                    colsample_bylevel=None,
                                    colsample_bynode=None,
                                    colsample_bytree=None, gamma=None,
                                    gpu_id=None, importance_type='g
                                    ain',
                                    interaction_constraints=None,
                                    learning_rate=None, max_delta_s
                                    tep=None,
                                    max_depth=None, min_child_weigh
                                    t=None,
                                    missing=nan, monotone_constrain
                                    ts=None,
                                    n_estimators=100, n_jobs...
                                    reg_alpha=None, reg_lambda=None,
                                    scale_pos_weight=None, silent=True,
                                    subsample=None, tree_method=None,
                                    validate_parameters=None, verbo
                                    sity=None),
            n_jobs=-1,
            param_grid={'colsample_bytree': [0.6], 'gamma': [4.0],
                        'learning_rate': [0.005, 0.01],
                        'max_depth': [12, 13, 14], 'min_child_weigh
                        t': [1],
                        'n_estimators': [1100], 'scale_pos_weight':
                        [1],
                        'subsample': [0.7]},
            scoring='roc_auc', verbose=2)
```

In [59] :

```
best_grid = grid_search.best_params_
best_grid
```

Out [59] :

```
{'colsample_bytree': 0.6,
'gamma': 4.0,
'learning_rate': 0.005,
'max_depth': 12,
'min_child_weight': 1,
'n_estimators': 1100,
'scale_pos_weight': 1,
'subsample': 0.7}
```

In [60] :

```
# Create a model with the parameters found
model_grid = xgb.XGBClassifier(objective='binary:logistic', silent=True,
                                 nthred=1, **best_grid)
fprs, tprs, scores = [], [], []
```

In [61]:

```
for (train, test), i in zip(cv.split(x, y), range(5)):
    model_grid.fit(x.iloc[train], y.iloc[train])
    _, _, auc_score_train = compute_roc_auc(model_grid, train)
    fpr, tpr, auc_score = compute_roc_auc(model_grid, test)
    scores.append((auc_score_train, auc_score))
    fprs.append(fpr)
    tprs.append(tpr)
```

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following:
g: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

```
[19:30:28] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "nthred", "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:30:28] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[19:30:57] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "nthred", "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:30:57] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[19:31:24] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "nthred", "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:31:24] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

```
[19:31:53] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:  
Parameters: { "nthred", "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```
[19:31:53] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the def
```

ault evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

[19:32:21] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:

Parameters: { "nthred", "silent" } might not be used.

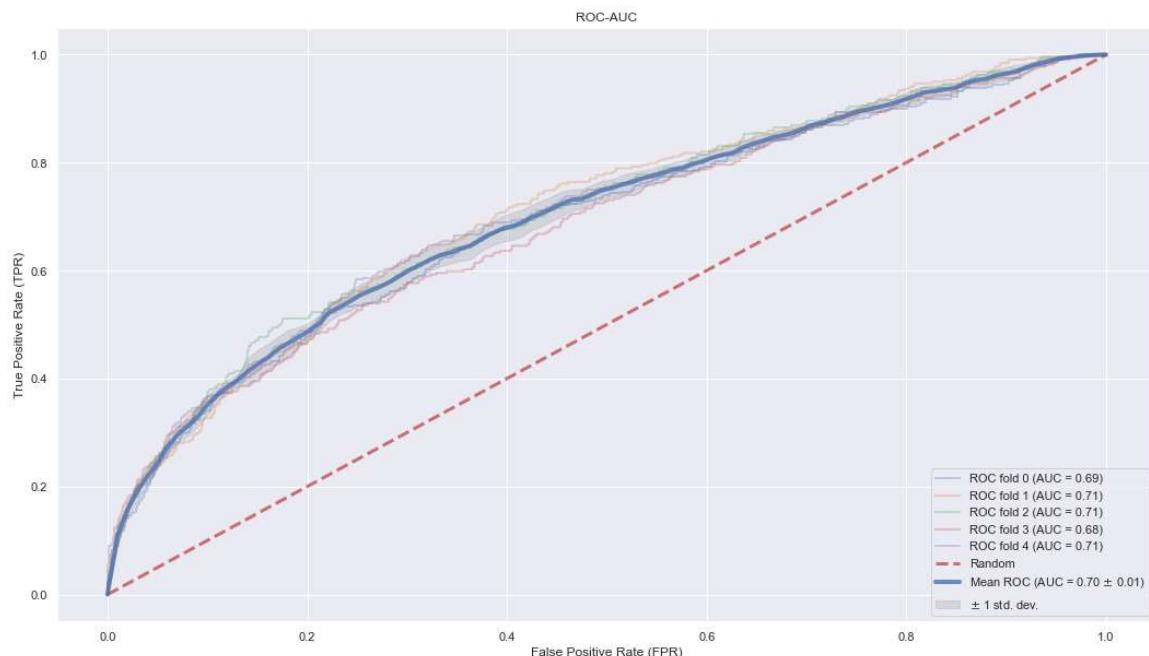
This may not be accurate due to some parameters are only used in language bindings but

passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

[19:32:21] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

In [62]:

```
plot_roc_curve(fprs, tprs)
plt.show()
```

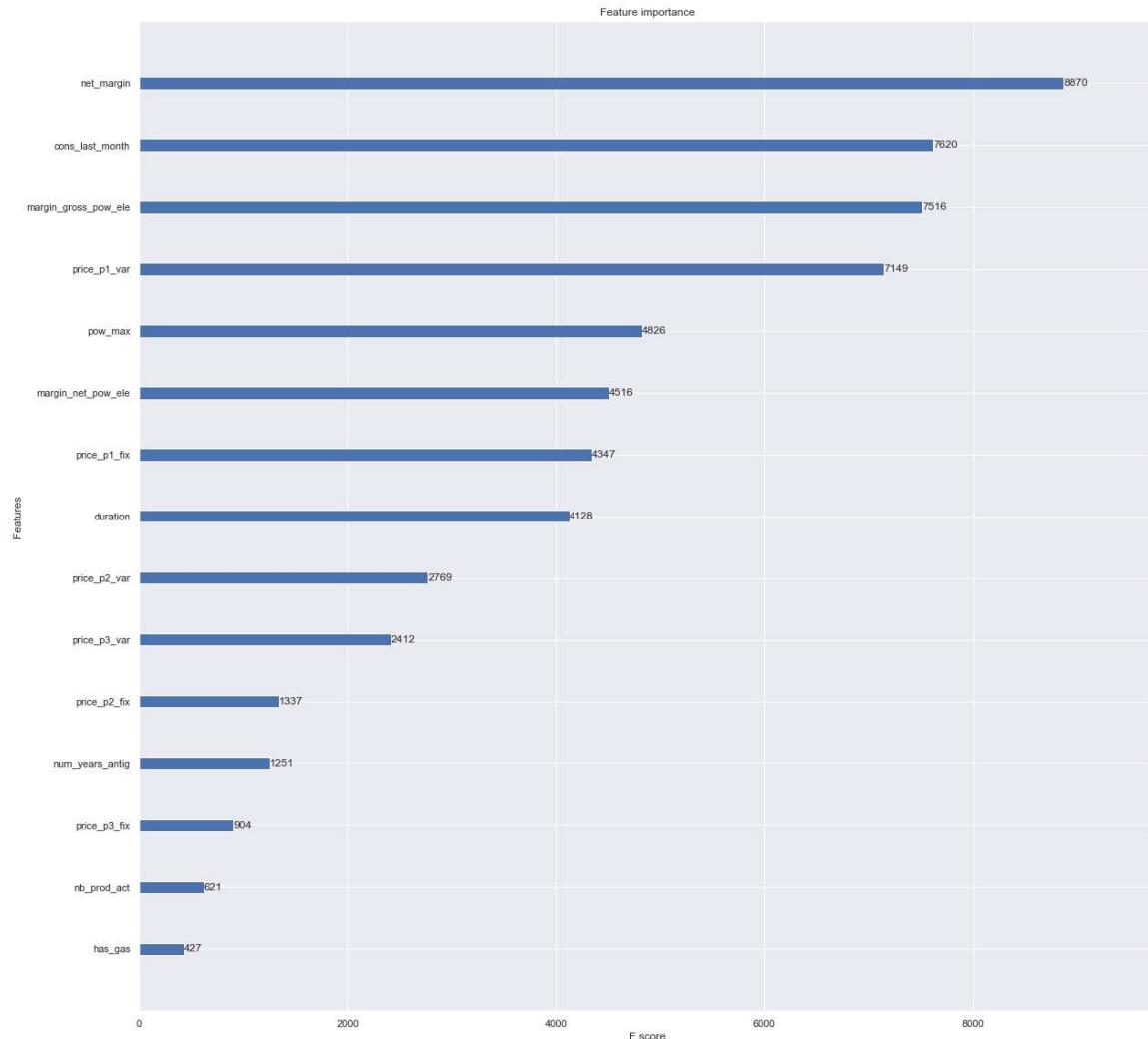


Understanding the model

One simple way of observing the feature importance is through counting the number of times each feature is split on across all boosting rounds (trees) in the model, and then visualizing the result as a bar graph, with the features ordered according to how many times they appear

In [63] :

```
fig, ax = plt.subplots(figsize=(20,20))
xgb.plot_importance(model_grid, ax=ax)
plt.show()
```



Partial dependence plot

In [64] :

```
from sklearn.inspection import plot_partial_dependence
```

currently there is a bug that does not allow us to use our trained model with pandas dataframes, we will create a replica and train it using numpy arrays

In [65]:

```
# Create a model with the parameters found
model_grid_v2 = xgb.XGBClassifier(objective='binary:logistic',
    silent=True, nthread=1, **best_grid)
model_grid_v2.fit(x_train.values,y_train.values)
```

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

```
[19:32:52] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:573:
Parameters: { "silent" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but

passed down to XGBoost core. Or some parameters are not used but slip through this

verification. Please open an issue if you find above cases.

```
[19:32:52] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out [65]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.6, gamma=4.0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.005, max_delta_step=0, max_depth=12,
              min_child_weight=1, missing=nan, monotone_constraints
              ='()', n_estimators=1100, n_jobs=1, nthread=1, num_parallel_trees=1,
              random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
              silent=True, subsample=0.7, tree_method='exact',
              validate_parameters=1, verbosity=None)
```

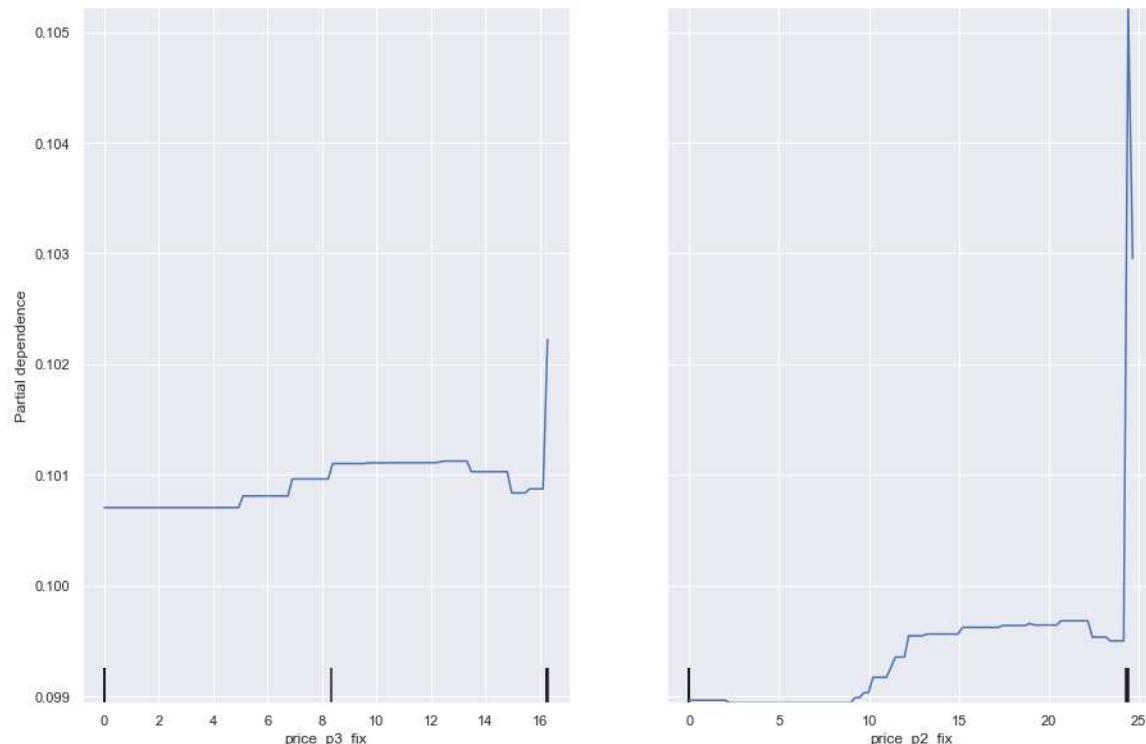
In [66]:

```
fig = plt.figure(figsize=(15,10))
plot_partial_dependence(model_grid_v2, x_test.values, features=[14, 13], feature_names=x_test.columns.tolist(), fig=fig)
```

The `fig` parameter is deprecated in version 0.22 and will be removed in version 0.24

Out [66]:

```
<sklearn.inspection._plot.partial_dependence.PartialDependenceDisplay at 0x272830f55c8>
```



SHAP- Feature Importance

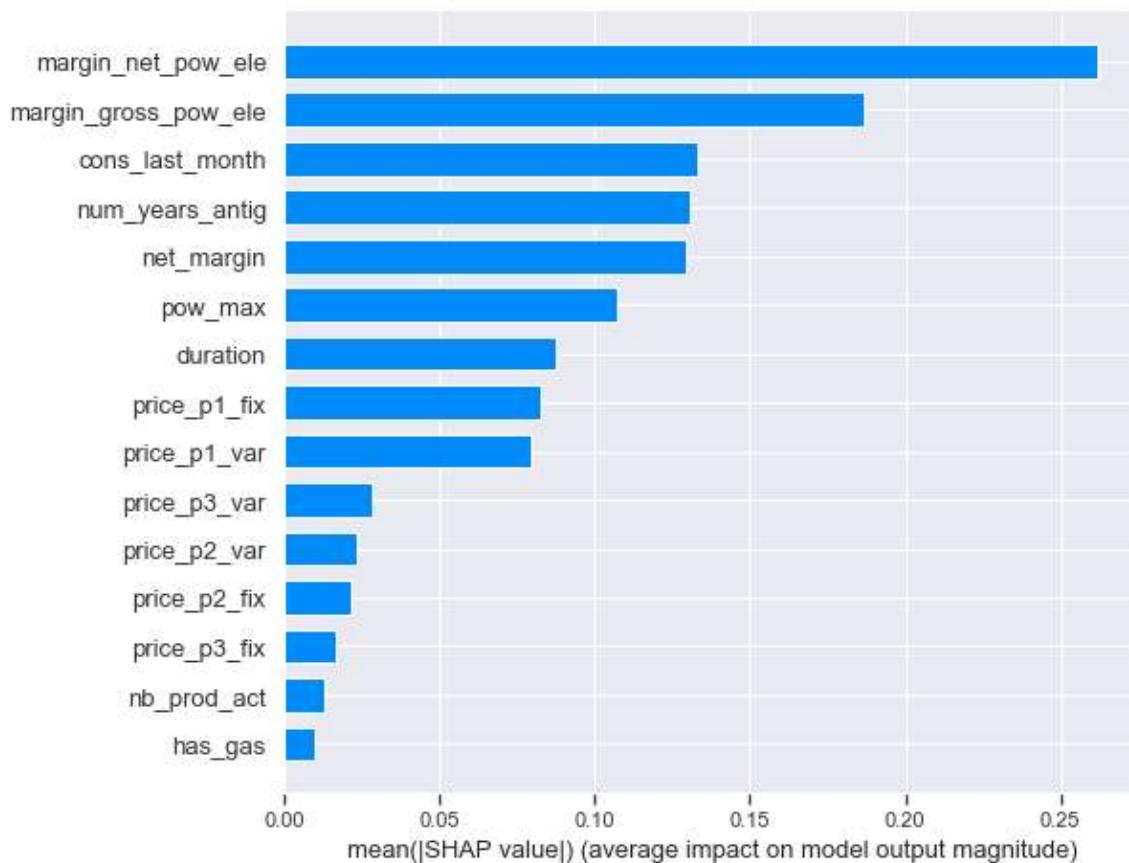
In [67]:

```
explainer = shap.TreeExplainer(model_grid)
shap_values = explainer.shap_values(x_test)
```

`ntree_limit` is deprecated, use `'iteration_range'` or model slicing instead.

In [68]:

```
# Feature importance for class 1 - churn  
shap.summary_plot(shap_values, x_test, plot_type='bar')
```



As expected the margin_net_pow_ele is the most important feature by far. It is interesting to compare how much important the top feature becomes in contrast with the other models we created Random forest and Logistic Regression