Homesite Quote Conversion

Pujan Malavia

```
In [ ]: from IPython.display import display
        from PIL import Image
        path= "C:/Users/puj83/OneDrive/Portfolio/Homesite_Quote_Conversion/homesite.pn
        g"
        display(Image.open(path))
```

Link to Dataset:

https://www.kaggle.com/c/homesite-quote-conversion/data (https://www.kaggle.com/c/homesite-quote-conversion/data)

Abstract:

Before asking someone on a date or skydiving, it's important to know your likelihood of success. The same goes for quoting home insurance prices to a potential customer. Homesite, a leading provider of homeowners insurance, does not currently have a dynamic conversion rate model that can give them confidence a quoted price will lead to a purchase.

Using an anonymized database of information on customer and sales activity, including property and coverage information, Homesite is challenging you to predict which customers will purchase a given quote. Accurately predicting conversion would help Homesite better understand the impact of proposed pricing changes and maintain an ideal portfolio of customer segments.

https://www.kaggle.com/c/homesite-quote-conversion (https://www.kaggle.com/c/homesite-quote-conversion)

Industry:

Insurance

Company Information:

Homesite. We've got you covered.

Founded in 1997, Homesite insurance was the first company to enable customers to purchase insurance directly online, during a single visit.

Since then, we’ve continued to innovate at the pace of our customers and their changing expectations. One thing that’s stayed the same since our founding: our commitment to our customers and partners.

We now offer Home, Renter, Life, Small Business, Condo and Flood Insurance.

A.M. Best has assigned an initial financial strength rating of A (Excellent) and an insurer credit rating of "A" to all Homesite Group insurance companies.

https://www.linkedin.com/company/homesite-insurance/about/ (https://www.linkedin.com/company/homesite-insurance/about/)

https://go.homesite.com/ (https://go.homesite.com/)

Use Case:

Predict which customers will purchase a given quote

Tool:

Python (Jupyter Notebook)

Initial Dataset:

train.csv

test.csv

Data:

This dataset represents the activity of a large number of customers who are interested in buying policies from Homesite. Each QuoteNumber corresponds to a potential customer and the QuoteConversion_Flag indicates whether the customer purchased a policy.

The provided features are anonymized and provide a rich representation of the prospective customer and policy. They include specific coverage information, sales information, personal information, property information, and geographic information. Your task is to predict QuoteConversion_Flag for each QuoteNumber in the test set.

Data Fields:

QuoteNumber

Original_Quote_Date

QuoteConversion_Flag

Field6

Field7 ....

Field11

Field12

CoverageField1A

CoverageField1B ....

CoverageField11A

CoverageField11B

SalesField1A

SalesField1B ....

SalesField14

SalesField15

PersonalField1

PersonalField2 ....

PersonalField83

PersonalField84

PropertyField1A

PropertyField1B ....

PropertyField39A

PropertyField39B

GeographicField1A

GeographicField1B ....

GeographicField63

GeographicField64

Import Libraries

In [ ]:
```
# !pip uninstall pandas
# !pip install pandas==0.19.2
# !pip install pandas-compat
# !pip install ggplot
# !pip uninstall pandas
# !pip install --user pandas==0.23.4
```

In [61]:
```
# import ggplot
import numpy as np
import pandas as pd
# import pandas_compat as pdc
from sklearn import preprocessing
import xgboost as xgb
import seaborn as sns
from pandas import Timestamp
from sklearn.preprocessing import LabelEncoder
# from pandas.lib import Timestamp
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split
from itertools import chain
```

Import Dataset(s)

In [62]:
```
# read data
train_file = 'C:/Users/puj83/OneDrive/Portfolio/Homesite_Quote_Conversion/trai
n.csv'
test_file = 'C:/Users/puj83/OneDrive/Portfolio/Homesite_Quote_Conversion/test.
csv'
sample_submission = 'C:/Users/puj83/OneDrive/Portfolio/Homesite_Quote_Conversi
on/sample_submission.csv'
```

In [3]:
```
xgb_params = {
    'seed': 0,
    'colsample_bytree': 0.8,
    'silent': 1,
    'subsample': .85,
    'eta': 0.0275,
    'objective': 'binary:logitraw',
    'num_parallel_tree': 7,
    'max_depth': 5,
    'nthread': 22,
    'eval_metric': 'auc',
}
```

In [4]:
```
top111 = ['Field12', 'PersonalField52', 'PersonalField80', 'PersonalField44', 'Field9',
          'PropertyField7', 'PropertyField12', 'CoverageField5B', 'PersonalField42', 'PersonalField45',
          'PersonalField81', 'PropertyField8', 'PersonalField79', 'GeographicField45B', 'PropertyField22',
          'PersonalField75', 'PersonalField31', 'PropertyField19', 'PropertyField31', 'GeographicField11A',
          'PersonalField23', 'GeographicField21B', 'PersonalField4A', 'Field10', 'GeographicField16B',
          'GeographicField20A', 'PersonalField25', 'PersonalField4B', 'PropertyField3', 'GeographicField17A',
          'GeographicField59B', 'GeographicField7B', 'GeographicField8A', 'Year', 'GeographicField6B',
          'PersonalField14',
          'GeographicField45A', 'GeographicField14B', 'SalesField12', 'CoverageField11A', 'CoverageField5A', 'Month',
          'PropertyField33', 'PersonalField5', 'CoverageField11B', 'GeographicField11B', 'GeographicField23B',
          'PropertyField39B', 'CoverageField3A', 'GeographicField1B', 'GeographicField17B', 'PropertyField39A',
          'GeographicField41B', 'CoverageField6A', 'SalesField9', 'PersonalField16', 'PersonalField26',
          'PropertyField24A', 'Field8', 'GeographicField28A', 'CoverageField3B', 'SalesField2A', 'GeographicField19B',
          'GeographicField43A', 'PropertyField16B', 'PropertyField16A', 'PropertyField1B', 'CoverageField1B',
          'PropertyField1A', 'GeographicField48B', 'PersonalField11', 'CoverageField1A', 'PersonalField15',
          'GeographicField5B', 'PropertyField34', 'CoverageField8', 'PersonalField82', 'SalesField2B',
          'PropertyField35', 'CoverageField2B', 'SalesField10', 'PropertyField21A', 'SalesField3', 'CoverageField9',
          'SalesField7', 'Weekday', 'PersonalField13', 'PropertyField21B', 'SalesField6', 'SalesField1A',
          'PersonalField9', 'SalesField4', 'PersonalField12', 'PersonalField27', 'PersonalField10B', 'Field7',
          'SalesField1B', 'PersonalField84', 'PersonalField2', 'PersonalField1', 'SalesField5', 'PersonalField10A',
          'PropertyField37', 'PropertyField29', 'GeographicField4B', 'PropertyField2B', 'GeographicField1A',
          'GeographicField61B', 'Field11', 'PersonalField76', 'PropertyField30']
```

In [5]:
```
drop_out = ['GeographicField19B', 'PropertyField7', 'GeographicField17A', 'Geo
graphicField28A',
            'GeographicField21B', 'GeographicField7B', 'CoverageField11B', 'Ge
ographicField6B', 'GeographicField45A',
            'PersonalField25', 'Month', 'CoverageField5A', 'GeographicField8A'
, 'GeographicField1B',
            'CoverageField6A_CoverageField6B', 'PersonalField23', 'Field11',
'PropertyField2B', 'SalesField12',
            'GeographicField41B',
            'PropertyField16A', 'Field10', 'PropertyField3', 'PropertyField16
B', 'GeographicField1A',
            'GeographicField20A', 'PersonalField81', 'GeographicField16B', 'Ge
ographicField59B', 'PersonalField79',
            'CoverageField1A_CoverageField3A', 'CoverageField3B_CoverageField4
B', 'PropertyField22',
            'GeographicField61B',
            'CoverageField3A_PropertyField21A', 'PropertyField12', 'CoverageFi
eld2A_CoverageField3A',
            'CoverageField2B_CoverageField3B', 'PropertyField8', 'PropertyFiel
d30', 'GeographicField14B',
            'PersonalField31',
            'PropertyField21A', 'CoverageField3A_CoverageField4A', 'PropertyFi
eld31', 'CoverageField11A',
            'PropertyField19', 'GeographicField45B', 'CoverageField1A', 'Perso
nalField75',
            'GeographicField8A_GeographicField13A', 'CoverageField3B_PropertyF
ield21B',
            'CoverageField1B_CoverageField3B', 'GeographicField6A_GeographicFi
eld13A', 'CoverageField5B',
            'PersonalField42', 'PersonalField45', 'PersonalField76', 'Geograph
icField6A_GeographicField8A',
            'PersonalField80', 'Field9', 'CoverageField3A', 'CoverageField3B',
            'GeographicField8A_GeographicField11A', 'GeographicField11A_Geogra
phicField13A',
            'GeographicField4B',
            'CoverageField2B', 'Field12', 'PropertyField21B', 'CoverageField1
B', 'PersonalField44',
            'GeographicField6A_GeographicField11A', 'PersonalField52']
```

In [6]:
```
interactions2way = [
    ("CoverageField1B", "PropertyField21B"),
    ("GeographicField6A", "GeographicField8A"),
    ("GeographicField6A", "GeographicField13A"),
    ("GeographicField8A", "GeographicField13A"),
    ("GeographicField11A", "GeographicField13A"),
    ("GeographicField8A", "GeographicField11A"),
    ("GeographicField6A", "GeographicField11A"),
    ("CoverageField6A", "CoverageField6B"),
    ("CoverageField3A", "CoverageField4A"),
    ("CoverageField2B", "CoverageField3B"),
    ("CoverageField1A", "CoverageField3A"),
    ("CoverageField3B", "CoverageField4B"),
    ("CoverageField2A", "CoverageField3A"),
    ("CoverageField1B", "CoverageField3B"),
    ("CoverageField3B", "PropertyField21B"),
    ("CoverageField3A", "PropertyField21A"),
    ("CoverageField1B", "PropertyField16B"),
    ("Weekday", "SalesField7"),
    ("PersonalField9", "CoverageField6B"),
    ("PersonalField12", "CoverageField6A"),
    ("PropertyField16B", "PropertyField21A"),
    ("PersonalField12", "Field8"),
    ("PropertyField32", "PersonalField9"),
    ("Field6", "CoverageField6A"),
    ("PersonalField12", "CoverageField6A"),
    ("CoverageField6A", "PropertyField34"),
    ("PersonalField33", "PropertyField8"),
    ("CoverageField2A", "CoverageField3B")
]
```

In [7]:
```
interactions3way = [('PersonalField23', 'PersonalField9', 'PropertyField37'),
                    ('CoverageField3A', 'PersonalField63', 'PropertyField21A'
),
                    ('CoverageField3A', 'CoverageField4A', 'PersonalField76'),
                    ('CoverageField3A', 'CoverageField4A', 'GeographicField62
A'),
                    ('CoverageField6A', 'PersonalField69', 'PersonalField9'),
                    ('CoverageField6A', 'PersonalField71', 'PersonalField9'),
                    ('GeographicField10B', 'GeographicField13A', 'PersonalFiel
d9'),
                    ('GeographicField8A', 'PersonalField71', 'PersonalField9'
),
                    ('CoverageField2B', 'PersonalField75', 'PropertyField16B'
),
                    ('CoverageField6A', 'PersonalField49', 'PropertyField29'),
                    ('CoverageField4B', 'PersonalField39', 'PropertyField16B'
),
                    ('CoverageField11B', 'PersonalField6', 'SalesField2B'),
                    ('CoverageField11B', 'PersonalField36', 'SalesField2B'),
                    ('CoverageField2B', 'PropertyField16B', 'PropertyField8'),
                    ('CoverageField3A', 'GeographicField21A', 'PropertyField21
B'),
                    ('GeographicField11A', 'PersonalField48', 'PersonalField9'
),
                    ('CoverageField11B', 'PersonalField26', 'SalesField2B'),
                    ('CoverageField1B', 'CoverageField3A', 'PersonalField61'),
                    ('CoverageField1A', 'PropertyField16A', 'PropertyField36'
),
                    ('PersonalField9', 'PropertyField10', 'PropertyField32'),
                    ('GeographicField11A', 'GeographicField62A', 'PersonalFiel
d12'),
                    ('Field10', 'PersonalField9', 'PropertyField34'),
                    ('CoverageField2B', 'CoverageField3A', 'PersonalField8'),
                    ('Field11', 'PropertyField34', 'SalesField6'),
                    ('PersonalField19', 'PersonalField60', 'PropertyField8')]
```

In [8]:
```
interactions4way = [('Field8', 'PersonalField12', 'PersonalField75', 'Property
Field37'),
                    ('CoverageField6A', 'PersonalField12', 'PropertyField37',
'PropertyField8'),
                    ('Field8', 'PersonalField9', 'PropertyField3', 'PropertyFi
eld37'),
                    ('CoverageField6A', 'Field8', 'PersonalField84', 'Personal
Field9'),
                    ('CoverageField8', 'PersonalField12', 'PersonalField80',
'PropertyField37'),
                    ('CoverageField8', 'Field8', 'PersonalField12', 'PersonalF
ield84'),
                    ('CoverageField5A', 'GeographicField11A', 'PersonalField9'
, 'PropertyField37'),
                    ('CoverageField1B', 'CoverageField3B', 'CoverageField5A',
'PropertyField22'),
                    ('CoverageField1A', 'CoverageField3A', 'PersonalField82',
'PropertyField19'),
                    ('CoverageField1A', 'CoverageField3A', 'PersonalField11',
'PropertyField19'),
                    ('CoverageField5A', 'Field8', 'PersonalField12', 'Personal
Field42'),
                    ('CoverageField6A', 'Field11', 'PersonalField9', 'Property
Field12'),
                    ('CoverageField6A', 'CoverageField8', 'PropertyField35',
'SalesField3'),
                    ('CoverageField3A', 'PersonalField82', 'PropertyField21A',
'Year'),
                    ('CoverageField1B', 'CoverageField3B', 'PersonalField42',
'PropertyField8'),
                    ('CoverageField1B', 'CoverageField3A', 'PersonalField1',
'PropertyField16A'),
                    ('CoverageField1B', 'CoverageField3B', 'PropertyField22',
'PropertyField8'),
                    ('CoverageField6A', 'PersonalField45', 'PersonalField9',
'PropertyField29'),
                    ('CoverageField5A', 'PersonalField1', 'PropertyField35',
'SalesField3'),
                    ('CoverageField1A', 'CoverageField3A', 'Field12', 'Persona
lField27'),
                    ('CoverageField5A', 'CoverageField8', 'Field11', 'Property
Field29'),
                    ('CoverageField3B', 'PersonalField25', 'PersonalField45',
'PropertyField21B'),
                    ('CoverageField2B', 'CoverageField3B', 'GeographicField17
A', 'PersonalField5'),
                    ('CoverageField1A', 'CoverageField3A', 'PersonalField75',
'Year'),
                    ('Field11', 'PersonalField12', 'PersonalField25', 'Propert
yField30')]
```

In [9]:
```
interactions2way_list = list(np.unique(list(chain(*interactions2way))))
interactions3way_list = list(np.unique(list(chain(*interactions3way))))
interactions4way_list = list(np.unique(list(chain(*interactions4way))))

interactions_list = interactions2way_list + interactions3way_list + interactio
ns4way_list
tmp_features = list(np.setdiff1d(interactions_list, top111))

tc_features = []
```

In [67]:
```
def get_data():
    global tc_features

    train = pd.read_csv(train_file)
    test = pd.read_csv(test_file)

    y_train = train.QuoteConversion_Flag

    train = train.drop(['QuoteNumber', 'QuoteConversion_Flag'], axis=1)
    test = test.drop('QuoteNumber', axis=1)

    ntrain = train.shape[0]

    train_test = pd.concat((train, test), axis=0)

    train_test['Date'] = pd.to_datetime(train_test['Original_Quote_Date'])

    train_test['Year'] = train_test['Date'].dt.year
    train_test['Month'] = train_test['Date'].dt.month
    train_test['Day'] = train_test['Date'].dt.day
    train_test['Weekday'] = train_test['Date'].dt.dayofweek

    train_test['Field10'] = train_test['Field10'].apply(lambda x: x.replace(
',', '')).astype(np.int32)
    train_test['PropertyField37'] = train_test['PropertyField37'].apply(lambda
x: -1 if x == ' ' else x)
    train_test['GeographicField63'] = train_test['GeographicField63'].apply(la
mbda x: -1 if x == ' ' else x)

    train_test = train_test.drop(['Date', 'Original_Quote_Date'], axis=1)
    train_test = train_test.fillna(-1)

    categoricals = [x for x in train_test.columns if train_test[x].dtype == 'o
bject']

    for c in categoricals:
        lbl = preprocessing.LabelEncoder()
        lbl.fit(list(train_test[c].values))
        train_test[c] = lbl.transform(list(train_test[c].values))

    train = train_test.iloc[:ntrain, :].copy().reset_index(drop=True)
    test = train_test.iloc[ntrain:, :].copy().reset_index(drop=True)

    features = list(train.columns)
    features = np.intersect1d(features, top111 + tmp_features)

    x_train = train[features].copy()
    x_test = test[features].copy()

    x_train['NaNCount'] = x_train.apply(lambda x: np.sum(x == -1), axis=1)
    x_test['NaNCount'] = x_test.apply(lambda x: np.sum(x == -1), axis=1)

    for A, B in interactions2way:
        feat = "_".join([A, B])
        x_train[feat] = x_train[A] - x_train[B]
        x_test[feat] = x_test[A] - x_test[B]
    for A, B, C in interactions3way:
        feat = "_".join([A, B, C])
        tc_features += [feat]
        x_train[feat] = x_train[A] - x_train[B] - x_train[C]
        x_test[feat] = x_test[A] - x_test[B] - x_test[C]

    for A, B, C, D in interactions4way:
        feat = "_".join([A, B, C, D])
        tc_features += [feat]
        x_train[feat] = x_train[A] - x_train[B] - x_train[C] - x_train[D]
        x_test[feat] = x_test[A] - x_test[B] - x_test[C] - x_test[D]

    x_train.drop(tmp_features, axis=1, inplace=True)
    x_test.drop(tmp_features, axis=1, inplace=True)

    x_train.drop(drop_out[-25:], axis=1, inplace=True)
    x_test.drop(drop_out[-25:], axis=1, inplace=True)

    return np.array(x_train), np.array(y_train), np.array(x_test)
```

In [68]:
```
if __name__ == "__main__":
    x_train, y_train, x_test = get_data()
    print (x_train.shape, x_test.shape)
```

```
(260753, 164) (173836, 164)
```

In [69]:
```
x_train_tc = x_train.copy()
ntcfeat = len(tc_features)

x_train[:, -ntcfeat:] = 0
ntrain = x_train.shape[0]
best_nrounds = 10

dtrain = xgb.DMatrix(x_train, label=y_train)
dtrain_tc = xgb.DMatrix(x_train_tc, label=y_train)
gbdt = xgb.train(xgb_params, dtrain, best_nrounds - 5)
xgb_params['eta'] = 0.01
gbdt = xgb.train(xgb_params, dtrain_tc, 10, xgb_model=gbdt)
dtest = xgb.DMatrix(x_test)

submission = pd.read_csv(sample_submission)
submission.iloc[:, 1] = gbdt1.predict(dtest).reshape((-1, 1))
submission.to_csv('C:/Users/puj83/OneDrive/Portfolio/Homesite_Quote_Conversio
n/submission.csv', index=False)
```

```
[17:33:05] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.
1.0\src\learner.cc:480:
Parameters: { silent } might not be used.

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


[17:33:24] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.
1.0\src\learner.cc:480:
Parameters: { silent } might not be used.

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