In [90]:
```
import numpy as np
import pandas as pd
```

In [91]:
```
fts = pd.read_excel('./data/features.xlsx', header=None)
fts = np.array(fts)
fts = [x[0].split()[1] for x in fts]
print(len(fts))
print(fts)
```

```
27
['pr_enrll_any', 'dep_bl_3am_svm', 'inv_bl_3am_svm', 'cr_bl_3am_svm', 'oprd_b
l_3am_svm', 'dep_oacc_ct_svm', 'ira_oacc_ct_svm', 'inv_oacc_ct_svm', 'meac_oa
cc_ct_svm', 'mesd_oacc_ct_svm', 'fsvc_oacc_ct_svm', 'cred_oacc_ct_svm', 'tma_
chnl_dc_ct_svm', 'tma_chnl_cc_ct_svm', 'tma_chnl_bcplt_ct_svm', 'tma_chnl_bct
lr_ct_svm', 'tma_chnl_atm_ct_svm', 'tma_chnl_olb_ct_svm', 'tma_chnl_mob_ct_sv
m', 'tma_chnl_ach_ct_svm', 'tma_chnl_icc_ct_svm', 'tma_chnl_dcc_ct_svm', 'tma
_chnl_mcc_ct_svm', 'tma_chnl_ccc_ct_svm', 'chnl_seg2_svm', 'prd_cat_svm', 'tt
l_cmp_svm']
```

In [92]:
```
fts_omp = [x.replace('svm', 'omp') for x in fts]
fts_tmp = [x.replace('svm', 'tmp') for x in fts]
fts_smp = [x.replace('svm', 'smp') for x in fts]
```

In [93]:
```
df = pd.read_csv('./data/customer_data.csv', header=0, index_col=0)
```

In [94]:
```
df_omp = df[fts_omp]
df_tmp = df[fts_tmp]
df_smp = df[fts_smp]
```

Modelling for 1 month prior information

In [95]:
```
df_omp.head()
```

Out[95]:

	pr_enrll_any	dep_bl_3am_omp	inv_bl_3am_omp	cr_bl_3am_omp	oprd_bl_3am_omp	dep_oac
0	N	10825.31	0.00	0.00	0	
1	N	22156.79	0.00	0.00	0	
2	N	37453.49	0.00	59.41	0	
3	N	64945.93	0.00	500.38	0	
4	N	5399.63	11284.51	0.00	0	

5 rows × 27 columns

In [96]:

```
# Label Encode Categorical Featrues
from sklearn import preprocessing

def clean(input_df):

    df = input_df.copy()

    # Label Encoding
    for col in cat:
        le = preprocessing.LabelEncoder()
        df[str(col)+'_Encoded'] = le.fit_transform(df[col].astype(str))

        le_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
        print(le_name_mapping)

        del df[col]

    return df
```

In [97]:

```
cat = list()
cat.append('chnl_seg2_omp')
cat.append('prd_cat_omp')

quant = list()
quant.append('dep_bl_3am_omp')
quant.append('inv_bl_3am_omp')
quant.append('cr_bl_3am_omp')
quant.append('tma_chnl_dc_ct_omp')
quant.append('tma_chnl_cc_ct_omp')
quant.append('tma_chnl_bcplt_ct_omp')
quant.append('tma_chnl_bctlr_ct_omp')
quant.append('tma_chnl_atm_ct_omp')
quant.append('tma_chnl_olb_ct_omp')
quant.append('tma_chnl_mob_ct_omp')
quant.append('tma_chnl_ach_ct_omp')

cleaned_omp = clean(df_omp)
cleaned_omp.head()
```

```
{'Mix': 0, 'No_Chnl': 1, 'PersonalTouch': 2, 'SelfService': 3}
{'crd_only': 0, 'dep_only': 1, 'inv_only': 2, 'multi_prd': 3, 'none': 4}
```

Out[97]:

	pr_enrll_any	dep_bl_3am_omp	inv_bl_3am_omp	cr_bl_3am_omp	oprd_bl_3am_omp	dep_oac
0	N	10825.31	0.00	0.00	0	
1	N	22156.79	0.00	0.00	0	
2	N	37453.49	0.00	59.41	0	
3	N	64945.93	0.00	500.38	0	
4	N	5399.63	11284.51	0.00	0	

5 rows × 27 columns

In [98]:

```
from sklearn.preprocessing import MinMaxScaler
from scipy.stats import skew

def preprocess(input_df,log,onehot):
    df = input_df.copy()

    # log transform the skewed features
    if log:

        #log transform skewed numeric features:
        skewed_feats = df[quant].apply(lambda x: skew(x)) #compute skewness
        skewed_feats = skewed_feats[skewed_feats > 0.75]
        skewed_feats = skewed_feats.index

        df[skewed_feats] = np.log1p(df[skewed_feats])

    # Convert to one-hot Encoding
    if onehot:
        encoded_features = [x + '_Encoded' for x in cat]

        onehotted = pd.get_dummies(data=df, columns=encoded_features)

        return onehotted
    else:
        return df
```

In [99]:

```
preprocessed_omp = preprocess(cleaned_omp, log=True, onehot=True).dropna()
preprocessed_omp.head()
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:15: RuntimeWarni
ng: invalid value encountered in log1p
  from ipykernel import kernelapp as app
```

Out[99]:

	pr_enrll_any	dep_bl_3am_omp	inv_bl_3am_omp	cr_bl_3am_omp	oprd_bl_3am_omp	dep_oac
0	N	9.289735	0.000000	0.000000	0	
1	N	10.005944	0.000000	0.000000	0	
2	N	10.530882	0.000000	4.101155	0	
3	N	11.081326	0.000000	6.217364	0	
4	N	8.594271	9.331275	0.000000	0	

5 rows × 34 columns

In [100]:

```
le = preprocessing.LabelEncoder()
preprocessed_omp['pr_enrll_any'] = le.fit_transform(preprocessed_omp['pr_enrll
_any'].astype(str))

y_omp = preprocessed_omp['pr_enrll_any']
X_omp = preprocessed_omp.drop(['pr_enrll_any'], axis=1)
```

In [101]:

```
from sklearn.linear_model import LassoCV

model_lasso = LassoCV(alphas = [0.0005, 0.005, 0.05, 0.5], n_jobs=4).fit(X_omp
, y_omp)

coef = pd.Series(model_lasso.coef_, index = X_omp.columns)
print("Lasso picked " + str(sum(coef != 0)) + " variables and eliminated the o
ther " +
        str(sum(coef == 0)) + " variables")
```

```
/usr/local/lib/python3.7/site-packages/sklearn/model_selection/_split.py:194
3: FutureWarning: You should specify a value for 'cv' instead of relying on t
he default value. The default value will change from 3 to 5 in version 0.22.
  warnings.warn(CV_WARNING, FutureWarning)

Lasso picked 27 variables and eliminated the other 6 variables
```

In [102]:

```
coef.sort_values()
```

Out[102]:

```
prd_cat_omp_Encoded_1     -0.686229
prd_cat_omp_Encoded_3     -0.651097
chnl_seg2_omp_Encoded_1   -0.083698
fsvc_oacc_ct_omp          -0.025255
tma_chnl_dc_ct_omp        -0.008731
cr_bl_3am_omp             -0.003155
ira_oacc_ct_omp           -0.000680
chnl_seg2_omp_Encoded_2   -0.000446
prd_cat_omp_Encoded_2      0.000000
meac_oacc_ct_omp           0.000000
inv_oacc_ct_omp            0.000000
oprd_bl_3am_omp            0.000000
prd_cat_omp_Encoded_0      0.000000
tma_chnl_icc_ct_omp        0.000000
tma_chnl_mcc_ct_omp        0.001259
tma_chnl_ccc_ct_omp        0.002040
mesd_oacc_ct_omp           0.004078
tma_chnl_dcc_ct_omp        0.004169
tma_chnl_ach_ct_omp        0.009784
tma_chnl_mob_ct_omp        0.010247
tma_chnl_atm_ct_omp        0.012654
chnl_seg2_omp_Encoded_3    0.013121
cred_oacc_ct_omp           0.014247
tma_chnl_cc_ct_omp         0.017936
inv_bl_3am_omp             0.018278
tma_chnl_olb_ct_omp        0.019047
dep_oacc_ct_omp            0.024871
chnl_seg2_omp_Encoded_0    0.027612
ttl_cmp_omp                0.028637
tma_chnl_bcplt_ct_omp      0.042347
tma_chnl_bctlr_ct_omp      0.046975
dep_bl_3am_omp             0.099015
prd_cat_omp_Encoded_4      0.688646
dtype: float64
```

Eliminated Variables

prd_cat_omp_Encoded_2 0.000000

meac_oacc_ct_omp 0.000000

inv_oacc_ct_omp 0.000000

oprd_bl_3am_omp 0.000000

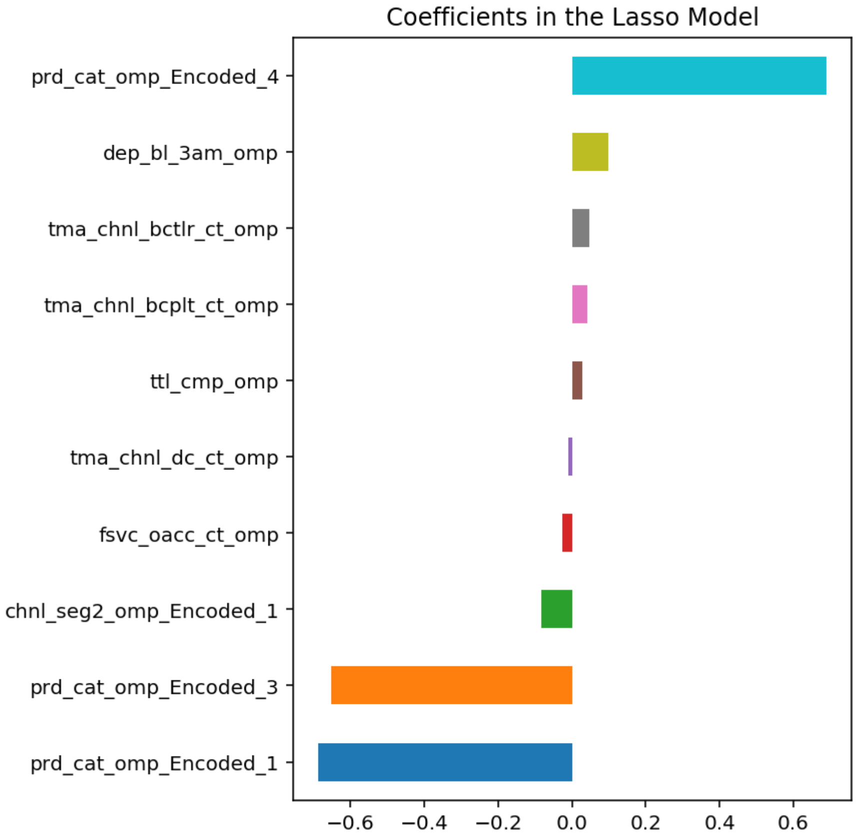
prd_cat_omp_Encoded_0 0.000000

tma_chnl_icc_ct_omp 0.000000

In [103]:

```
import matplotlib
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
%matplotlib inline

imp_coef = pd.concat([coef.sort_values().head(5),
                     coef.sort_values().tail(5)])
matplotlib.rcParams['figure.figsize'] = (5, 7)
imp_coef.plot(kind = "barh")
a = plt.title("Coefficients in the Lasso Model")
```



Modelling for 3 month prior information

In [104]:

```
df_tmp.head()
```

Out[104]:

	pr_enrll_any	dep_bl_3am_tmp	inv_bl_3am_tmp	cr_bl_3am_tmp	oprd_bl_3am_tmp	dep_oacc_
0	N	12107.96	0.00	0.00	0	
1	N	21459.42	0.00	0.00	0	
2	N	37769.06	0.00	75.78	0	
3	N	44071.71	0.00	55.00	0	
4	N	3347.64	12149.44	0.00	0	

5 rows × 27 columns

In [105]:

```
cat = list()
cat.append('chnl_seg2_tmp')
cat.append('prd_cat_tmp')

quant = list()
quant.append('dep_bl_3am_tmp')
quant.append('inv_bl_3am_tmp')
quant.append('cr_bl_3am_tmp')
quant.append('tma_chnl_dc_ct_tmp')
quant.append('tma_chnl_cc_ct_tmp')
quant.append('tma_chnl_bcplt_ct_tmp')
quant.append('tma_chnl_bctlr_ct_tmp')
quant.append('tma_chnl_atm_ct_tmp')
quant.append('tma_chnl_olb_ct_tmp')
quant.append('tma_chnl_mob_ct_tmp')
quant.append('tma_chnl_ach_ct_tmp')

cleaned_tmp = clean(df_tmp)
cleaned_tmp.head()
```

```
{'Mix': 0, 'No_Chnl': 1, 'PersonalTouch': 2, 'SelfService': 3}
{'crd_only': 0, 'dep_only': 1, 'inv_only': 2, 'multi_prd': 3, 'none': 4}
```

Out[105]:

	pr_enrll_any	dep_bl_3am_tmp	inv_bl_3am_tmp	cr_bl_3am_tmp	oprd_bl_3am_tmp	dep_oacc_
0	N	12107.96	0.00	0.00	0	
1	N	21459.42	0.00	0.00	0	
2	N	37769.06	0.00	75.78	0	
3	N	44071.71	0.00	55.00	0	
4	N	3347.64	12149.44	0.00	0	

5 rows × 27 columns

In [106]:
```
preprocessed_tmp = preprocess(cleaned_tmp, log=True, onehot=True).dropna()
preprocessed_tmp.head()
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:15: RuntimeWarni
ng: invalid value encountered in log1p
  from ipykernel import kernelapp as app
```

Out[106]:

	pr_enrll_any	dep_bl_3am_tmp	inv_bl_3am_tmp	cr_bl_3am_tmp	oprd_bl_3am_tmp	dep_oacc_
0	N	9.401701	0.000000	0.000000	0	
1	N	9.973966	0.000000	0.000000	0	
2	N	10.539272	0.000000	4.340944	0	
3	N	10.693596	0.000000	4.025352	0	
4	N	8.116310	9.405121	0.000000	0	

5 rows × 34 columns

In [107]:
```
le = preprocessing.LabelEncoder()
preprocessed_tmp['pr_enrll_any'] = le.fit_transform(preprocessed_tmp['pr_enrll
_any'].astype(str))

y_tmp = preprocessed_tmp['pr_enrll_any']
X_tmp = preprocessed_tmp.drop(['pr_enrll_any'], axis=1)
```

In [108]:
```
model_lasso = LassoCV(alphas = [0.0005, 0.005, 0.05, 0.5], n_jobs=4).fit(X_tmp
, y_tmp)

coef = pd.Series(model_lasso.coef_, index = X_tmp.columns)
print("Lasso picked " + str(sum(coef != 0)) + " variables and eliminated the o
ther " +
        str(sum(coef == 0)) + " variables")
```

```
/usr/local/lib/python3.7/site-packages/sklearn/model_selection/_split.py:194
3: FutureWarning: You should specify a value for 'cv' instead of relying on t
he default value. The default value will change from 3 to 5 in version 0.22.
  warnings.warn(CV_WARNING, FutureWarning)

Lasso picked 28 variables and eliminated the other 5 variables
```

```
In [109]: coef.sort_values()
```

```
Out[109]: prd_cat_tmp_Encoded_1     -0.479494
          prd_cat_tmp_Encoded_3     -0.450807
          chnl_seg2_tmp_Encoded_1   -0.081246
          fsvc_oacc_ct_tmp          -0.013956
          tma_chnl_dc_ct_tmp        -0.009666
          cr_bl_3am_tmp             -0.002539
          ira_oacc_ct_tmp           -0.001077
          chnl_seg2_tmp_Encoded_2   -0.000105
          prd_cat_tmp_Encoded_2      0.000000
          oprd_bl_3am_tmp            0.000000
          prd_cat_tmp_Encoded_0      0.000000
          inv_oacc_ct_tmp           -0.000000
          meac_oacc_ct_tmp          -0.000000
          ttl_cmp_tmp                0.000965
          tma_chnl_icc_ct_tmp        0.002003
          tma_chnl_mcc_ct_tmp        0.002632
          tma_chnl_dcc_ct_tmp        0.004910
          tma_chnl_ccc_ct_tmp        0.005654
          tma_chnl_mob_ct_tmp        0.008939
          tma_chnl_ach_ct_tmp        0.010173
          inv_bl_3am_tmp             0.011800
          cred_oacc_ct_tmp           0.015281
          tma_chnl_atm_ct_tmp        0.016666
          tma_chnl_cc_ct_tmp         0.016918
          chnl_seg2_tmp_Encoded_3    0.018528
          tma_chnl_olb_ct_tmp        0.019210
          mesd_oacc_ct_tmp           0.020279
          chnl_seg2_tmp_Encoded_0    0.021326
          dep_oacc_ct_tmp            0.025829
          tma_chnl_bcplt_ct_tmp      0.033689
          tma_chnl_bctlr_ct_tmp      0.050384
          dep_bl_3am_tmp             0.072755
          prd_cat_tmp_Encoded_4      0.815210
          dtype: float64
```

Eliminated Variables

prd_cat_tmp_Encoded_2 0.000000

oprd_bl_3am_tmp 0.000000

prd_cat_tmp_Encoded_0 0.000000

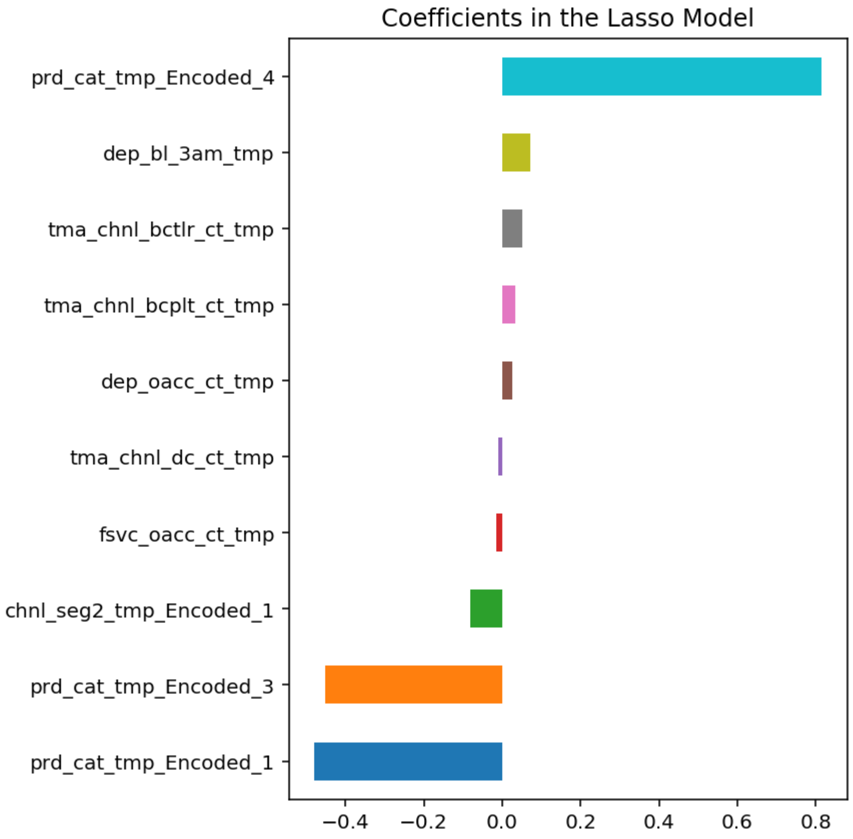
inv_oacc_ct_tmp -0.000000

meac_oacc_ct_tmp -0.000000

In [110]:

```
import matplotlib
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
%matplotlib inline

imp_coef = pd.concat([coef.sort_values().head(5),
                     coef.sort_values().tail(5)])
matplotlib.rcParams['figure.figsize'] = (5, 7)
imp_coef.plot(kind = "barh")
a = plt.title("Coefficients in the Lasso Model")
```



Modelling for 6 month prior information

In [111]:
```
df_smp.head()
```

Out[111]:

	pr_enrll_any	dep_bl_3am_smp	inv_bl_3am_smp	cr_bl_3am_smp	oprd_bl_3am_smp	dep_oac
0	N	10258.23	0.00	0.00	0	
1	N	18021.82	0.00	0.00	0	
2	N	37137.98	0.00	36.15	0	
3	N	127342.55	0.00	0.00	0	
4	N	791.01	19432.17	0.00	0	

5 rows × 27 columns

In [112]:
```
cat = list()
cat.append('chnl_seg2_smp')
cat.append('prd_cat_smp')

quant = list()
quant.append('dep_bl_3am_smp')
quant.append('inv_bl_3am_smp')
quant.append('cr_bl_3am_smp')
quant.append('tma_chnl_dc_ct_smp')
quant.append('tma_chnl_cc_ct_smp')
quant.append('tma_chnl_bcplt_ct_smp')
quant.append('tma_chnl_bctlr_ct_smp')
quant.append('tma_chnl_atm_ct_smp')
quant.append('tma_chnl_olb_ct_smp')
quant.append('tma_chnl_mob_ct_smp')
quant.append('tma_chnl_ach_ct_smp')

cleaned_smp = clean(df_smp)
cleaned_smp.head()
```

```
{'Mix': 0, 'No_Chnl': 1, 'PersonalTouch': 2, 'SelfService': 3}
{'crd_only': 0, 'dep_only': 1, 'inv_only': 2, 'multi_prd': 3, 'none': 4}
```

Out[112]:

	pr_enrll_any	dep_bl_3am_smp	inv_bl_3am_smp	cr_bl_3am_smp	oprd_bl_3am_smp	dep_oac
0	N	10258.23	0.00	0.00	0	
1	N	18021.82	0.00	0.00	0	
2	N	37137.98	0.00	36.15	0	
3	N	127342.55	0.00	0.00	0	
4	N	791.01	19432.17	0.00	0	

5 rows × 27 columns

In [113]:

```
preprocessed_smp = preprocess(cleaned_smp, log=True, onehot=True).dropna()
preprocessed_smp.head()
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:15: RuntimeWarni
ng: invalid value encountered in log1p
  from ipykernel import kernelapp as app
```

Out[113]:

	pr_enrll_any	dep_bl_3am_smp	inv_bl_3am_smp	cr_bl_3am_smp	oprd_bl_3am_smp	dep_oac
0	N	9.235933	0.000000	0.000000	0	
1	N	9.799394	0.000000	0.000000	0	
2	N	10.522422	0.000000	3.614964	0	
3	N	11.754644	0.000000	0.000000	0	
4	N	6.674574	9.874737	0.000000	0	

5 rows × 34 columns

In [114]:

```
le = preprocessing.LabelEncoder()
preprocessed_smp['pr_enrll_any'] = le.fit_transform(preprocessed_smp['pr_enrll
_any'].astype(str))

y_smp = preprocessed_smp['pr_enrll_any']
X_smp = preprocessed_smp.drop(['pr_enrll_any'], axis=1)
```

In [115]:

```
model_lasso = LassoCV(alphas = [0.0005, 0.005, 0.05, 0.5], n_jobs=4).fit(X_smp
, y_smp)

coef = pd.Series(model_lasso.coef_, index = X_smp.columns)
print("Lasso picked " + str(sum(coef != 0)) + " variables and eliminated the o
ther " +
        str(sum(coef == 0)) + " variables")
```

```
/usr/local/lib/python3.7/site-packages/sklearn/model_selection/_split.py:194
3: FutureWarning: You should specify a value for 'cv' instead of relying on t
he default value. The default value will change from 3 to 5 in version 0.22.
  warnings.warn(CV_WARNING, FutureWarning)

Lasso picked 28 variables and eliminated the other 5 variables
```

```
In [116]: coef.sort_values()
```

```
Out[116]: prd_cat_smp_Encoded_1     -0.311420
          prd_cat_smp_Encoded_3     -0.286942
          chnl_seg2_smp_Encoded_1   -0.093675
          tma_chnl_dc_ct_smp        -0.010513
          fsvc_oacc_ct_smp          -0.007167
          cr_bl_3am_smp             -0.002426
          tma_chnl_icc_ct_smp       -0.002199
          ira_oacc_ct_smp           -0.001152
          tma_chnl_mcc_ct_smp       -0.000290
          chnl_seg2_smp_Encoded_2   -0.000174
          prd_cat_smp_Encoded_2      0.000000
          oprd_bl_3am_smp            0.000000
          ttl_cmp_smp                0.000000
          meac_oacc_ct_smp          -0.000000
          inv_oacc_ct_smp           -0.000000
          tma_chnl_ccc_ct_smp        0.003364
          tma_chnl_dcc_ct_smp        0.004589
          inv_bl_3am_smp             0.005947
          tma_chnl_mob_ct_smp        0.007178
          tma_chnl_ach_ct_smp        0.011524
          chnl_seg2_smp_Encoded_3    0.011741
          cred_oacc_ct_smp           0.015603
          tma_chnl_atm_ct_smp        0.016533
          tma_chnl_cc_ct_smp         0.016876
          tma_chnl_olb_ct_smp        0.018815
          chnl_seg2_smp_Encoded_0    0.022190
          tma_chnl_bcplt_ct_smp      0.026108
          dep_oacc_ct_smp            0.027281
          mesd_oacc_ct_smp           0.031852
          tma_chnl_bctlr_ct_smp      0.048220
          dep_bl_3am_smp             0.048841
          prd_cat_smp_Encoded_0      0.454731
          prd_cat_smp_Encoded_4      0.633309
          dtype: float64
```

Eliminated Variables

prd_cat_smp_Encoded_2 0.000000

oprd_bl_3am_smp 0.000000

ttl_cmp_smp 0.000000

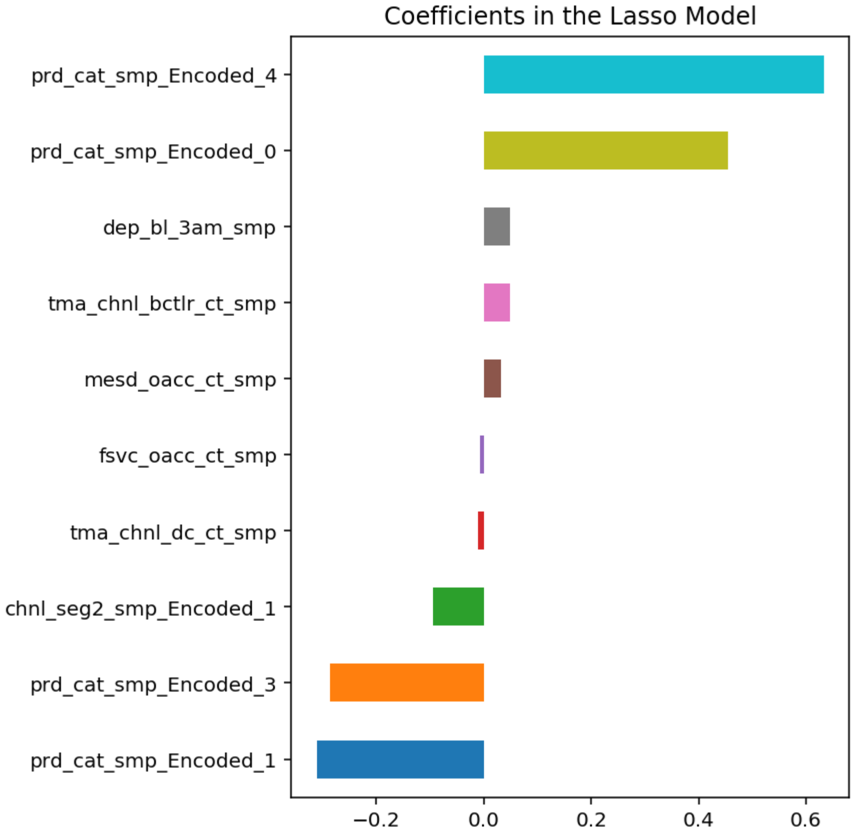
meac_oacc_ct_smp -0.000000

inv_oacc_ct_smp -0.000000

In [117]:

```
import matplotlib
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
%matplotlib inline

imp_coef = pd.concat([coef.sort_values().head(5),
                     coef.sort_values().tail(5)])
matplotlib.rcParams['figure.figsize'] = (5, 7)
imp_coef.plot(kind = "barh")
a = plt.title("Coefficients in the Lasso Model")
```



In [123]:

```
print(fts_smp)
```

```
['pr_enrll_any', 'dep_bl_3am_smp', 'inv_bl_3am_smp', 'cr_bl_3am_smp', 'oprd_b
l_3am_smp', 'dep_oacc_ct_smp', 'ira_oacc_ct_smp', 'inv_oacc_ct_smp', 'meac_oa
cc_ct_smp', 'mesd_oacc_ct_smp', 'fsvc_oacc_ct_smp', 'cred_oacc_ct_smp', 'tma_
chnl_dc_ct_smp', 'tma_chnl_cc_ct_smp', 'tma_chnl_bcplt_ct_smp', 'tma_chnl_bct
lr_ct_smp', 'tma_chnl_atm_ct_smp', 'tma_chnl_olb_ct_smp', 'tma_chnl_mob_ct_sm
p', 'tma_chnl_ach_ct_smp', 'tma_chnl_icc_ct_smp', 'tma_chnl_dcc_ct_smp', 'tma
_chnl_mcc_ct_smp', 'tma_chnl_ccc_ct_smp', 'chnl_seg2_smp', 'prd_cat_smp', 'tt
l_cmp_smp']
```

In [ ]: