In [10]:

```
precision_score(y_test, y_pred)
```

Out[10]:

```
0.5297728667894414
```

In [11]:

```
recall_score(y_test, y_pred)
```

Out[11]:

```
0.6732544532570537
```

In [12]:

```
accuracy_score(y_test, y_pred)
```

Out[12]:

```
0.6912218216566043
```

In [13]:

```
fpr, tpr, thresholds = roc_curve(y_true=y_test, y_score=y_prob)
AUROC = roc_auc_score(y_true=y_test, y_score=y_prob)*100
```

In [14]:

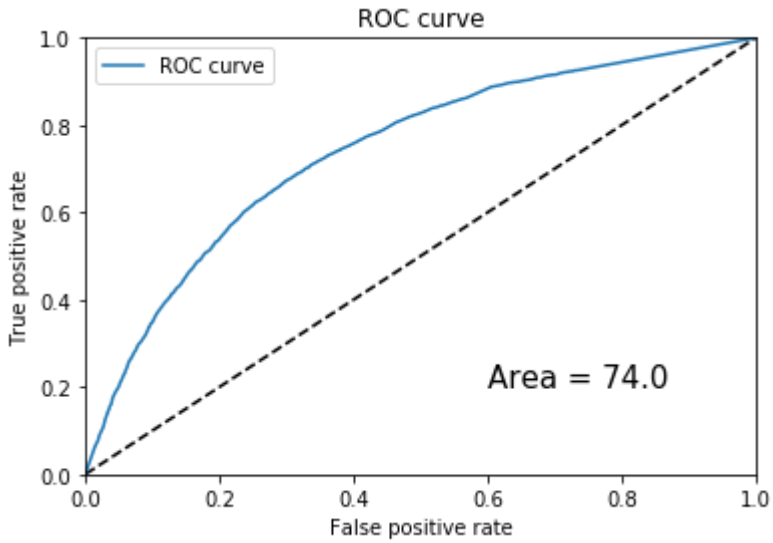
```
roc_auc_score(y_true=y_test, y_score=y_prob)
```

Out[14]:

```
0.7399985964448028
```

In [15]:

```
plt.figure(1)
plt.plot([0, 1], [0, 1], 'k--')
plt.plot(fpr, tpr, label='ROC curve')

plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')

plt.xlim(0, 1)
plt.ylim(0, 1)

plt.text(0.6, 0.2, "Area = {}".format(round(AUROC,2)), dict(size=15))

plt.legend(loc='best')
plt.show()
```



In [16]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       False       0.81      0.70      0.75     30664
        True       0.53      0.67      0.59     15382

   micro avg       0.69      0.69      0.69     46046
   macro avg       0.67      0.69      0.67     46046
weighted avg       0.72      0.69      0.70     46046
```

In [17]:

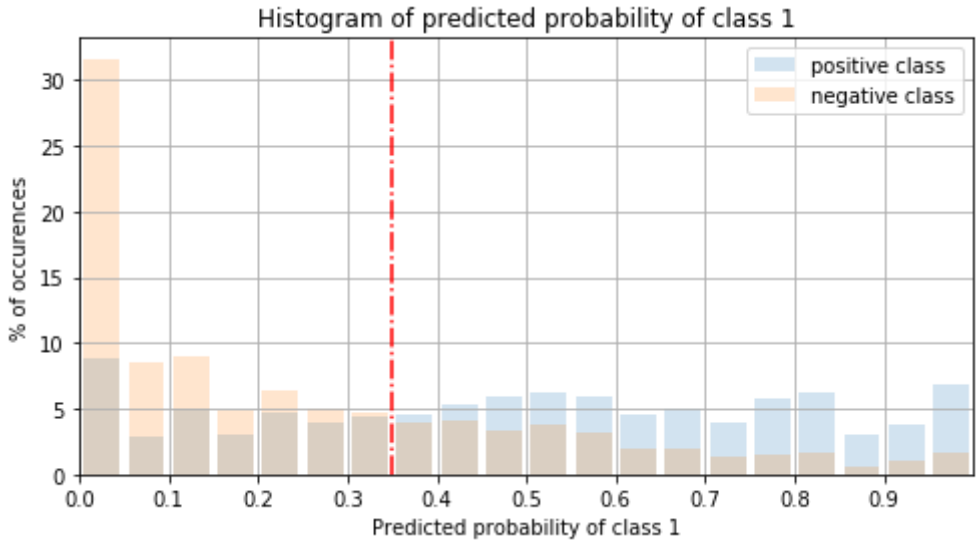
```
df_results = y_test.to_frame(name='y_true')
df_results['y_pred'] = y_pred
df_results['y_prob'] = y_prob
df_results.head()
```

Out[17]:

	y_true	y_pred	y_prob
30717	False	False	0.183333
44889	True	True	0.828644
13938	False	False	0.206627
20220	False	False	0.265000
21837	True	True	0.700000

In [18]:

```
n_trues = df_results[df_results.y_true==True].shape[0]
n_falses = df_results[df_results.y_true==False].shape[0]

plt.figure(figsize=(8,4))
df_results[df_results.y_true==True].y_prob.hist(bins=20, weights=np.ones(n_trues)/n_tru
es*100, align='mid', rwidth=0.8, alpha=0.2, label='positive class')
df_results[df_results.y_true==False].y_prob.hist(bins=20, weights=np.ones(n_falses)/n_f
alses*100, align='mid', rwidth=0.8, alpha=0.2, label='negative class')

plt.axvline(x=decision_thr, color='r', linestyle='-.')

plt.title('Histogram of predicted probability of class 1')
plt.xlabel("Predicted probability of class 1")
plt.ylabel("% of occurences")

plt.xlim(0,1)
plt.xticks(np.arange(0, 1, step=0.1))

plt.legend();
```



In [19]:

```
df_results[df_results.y_true==True].y_prob.hist(bins=20, figsize=(8,4), histtype='step'
,label='positive class', density=True, cumulative=True)
df_results[df_results.y_true==False].y_prob.hist(bins=20, figsize=(8,4), histtype='ste
p', label='negative class', density=True, cumulative=True)

plt.axvline(x=decision_thr, color='r', linestyle='-.')

plt.title('Histogram of predicted probability of class 1')
plt.xlabel("Predicted probability of class 1")
plt.ylabel("Density")

plt.xlim(0,1)
plt.xticks(np.arange(0, 1, step=0.1))

plt.legend();
```