Online Supplemental Materials

Section 1: Preliminary Analyses Examining Relationships Between Demographic Variables and Academic Outcomes and Possible Identity Scores

Table S1. 
Correlation between child and school level measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FRPL (School-Level)</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. FRPL (Child-level)</td>
<td>0.188**</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Latinx (School-Level)</td>
<td>0.072</td>
<td>-0.051</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Latinx (Child-level)</td>
<td>0.098</td>
<td>-0.009</td>
<td>0.763**</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Black (School-Level)</td>
<td>0.020</td>
<td>0.070</td>
<td>-0.995**</td>
<td>-0.758**</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Black (Child-level)</td>
<td>0.002</td>
<td>0.074</td>
<td>-0.824**</td>
<td>-0.898**</td>
<td>0.829**</td>
<td>--</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Female (Student-Level)</td>
<td>-0.043</td>
<td>0.045</td>
<td>-0.044</td>
<td>-0.084</td>
<td>0.042</td>
<td>0.110</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>8. Student-Teacher Ratio</td>
<td>-0.063</td>
<td>-0.018</td>
<td>0.410**</td>
<td>0.268**</td>
<td>-0.378**</td>
<td>-0.300**</td>
<td>0.008</td>
<td>--</td>
</tr>
</tbody>
</table>

Note: **p<.01; FRPL = Free or reduced-price lunch status
### Table S2.
*Correlation between child and school level measures and GPA*

<table>
<thead>
<tr>
<th></th>
<th>FRPL (School-Level)</th>
<th>FRPL (Child-level)</th>
<th>Latinx (School-Level)</th>
<th>Latinx (Child-level)</th>
<th>Female (Student-Level)</th>
<th>Student-Teacher Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core GPA 6th</td>
<td>-0.263**</td>
<td>-0.143*</td>
<td>-0.076</td>
<td>-0.073</td>
<td>0.280**</td>
<td>0.182**</td>
</tr>
<tr>
<td>Core GPA 7th</td>
<td>-0.095</td>
<td>-0.111</td>
<td>-0.176**</td>
<td>-0.141*</td>
<td>0.284**</td>
<td>-0.046</td>
</tr>
<tr>
<td>Core GPA 8th</td>
<td>-0.224**</td>
<td>-0.155*</td>
<td>-0.050</td>
<td>-0.053</td>
<td>0.299**</td>
<td>0.089</td>
</tr>
<tr>
<td>6th-7th GPA Change</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7th-8th GPA Change</td>
<td>0.280**</td>
<td>0.050</td>
<td>-0.174**</td>
<td>-0.118</td>
<td>0.015</td>
<td>-0.386**</td>
</tr>
<tr>
<td>Change</td>
<td>-0.165**</td>
<td>-0.029</td>
<td>0.255**</td>
<td>0.183**</td>
<td>-0.075</td>
<td>0.223**</td>
</tr>
</tbody>
</table>

*Note: * p<.01, ** p<.01; FRPL = Free or reduced-price lunch status*

### Table S3.
*Correlation between child and school level measures and possible identity scores*

<table>
<thead>
<tr>
<th></th>
<th>FRPL (School-Level)</th>
<th>FRPL (Child-level)</th>
<th>Latinx (School-Level)</th>
<th>Latinx (Child-level)</th>
<th>Female (Student-Level)</th>
<th>Student-Teacher Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall School-Focused PI Count</td>
<td>0.111</td>
<td>0.047</td>
<td>-0.024</td>
<td>-0.009</td>
<td>-0.032</td>
<td>0.011</td>
</tr>
<tr>
<td>Fall School-Focused PI Balance Count</td>
<td>0.058</td>
<td>0.038</td>
<td>0.008</td>
<td>0.037</td>
<td>-0.003</td>
<td>0.022</td>
</tr>
<tr>
<td>Fall School-Focused PI with Strategies Count</td>
<td>0.081</td>
<td>0.021</td>
<td>-0.031</td>
<td>-0.048</td>
<td>-0.064</td>
<td>0.001</td>
</tr>
<tr>
<td>Fall School-Focused PI Plausibility</td>
<td>0.064</td>
<td>0.085</td>
<td>-0.031</td>
<td>-0.038</td>
<td>0.029</td>
<td>0.076</td>
</tr>
<tr>
<td>Spring School-Focused PI Count</td>
<td>0.081</td>
<td>-0.032</td>
<td>0.061</td>
<td>0.120</td>
<td>0.093</td>
<td>-0.082</td>
</tr>
<tr>
<td>Spring School-Focused PI Balance Count</td>
<td>0.103</td>
<td>0.062</td>
<td>0.048</td>
<td>0.089</td>
<td>0.128*</td>
<td>-0.008</td>
</tr>
<tr>
<td>Spring School-Focused PI with Strategies Count</td>
<td>0.053</td>
<td>-0.025</td>
<td>-0.019</td>
<td>0.050</td>
<td>0.067</td>
<td>-0.116</td>
</tr>
<tr>
<td>Spring School-Focused PI Plausibility</td>
<td>0.083</td>
<td>-0.011</td>
<td>0.048</td>
<td>0.099</td>
<td>0.152*</td>
<td>0.006</td>
</tr>
</tbody>
</table>
WHICH CHANGES IN POSSIBLE SELVES MATTER?

<table>
<thead>
<tr>
<th>School-Focused PI Count</th>
<th>Change</th>
<th>0.059</th>
<th>-0.043</th>
<th>0.068</th>
<th>0.125*</th>
<th>0.102</th>
<th>-0.087</th>
</tr>
</thead>
<tbody>
<tr>
<td>School-Focused PI Balance Count Change</td>
<td>0.092</td>
<td>0.054</td>
<td>0.047</td>
<td>0.083</td>
<td>0.133*</td>
<td>-0.014</td>
<td></td>
</tr>
<tr>
<td>School-Focused PI with Strategies Count Change</td>
<td>0.037</td>
<td>-0.03</td>
<td>-0.013</td>
<td>0.062</td>
<td>0.082</td>
<td>-0.119</td>
<td></td>
</tr>
<tr>
<td>School-Focused PI Plausibility Change</td>
<td>0.072</td>
<td>-0.029</td>
<td>0.055</td>
<td>0.109</td>
<td>0.149*</td>
<td>-0.009</td>
<td></td>
</tr>
</tbody>
</table>

*Note: PI=possible identity*

Table S4 Post-hoc sensitivity analyses for Models 1, 3, 4 (Table 5) with n=247: Possible Identity Change Scores Predicting End-of-Year GPA

**Model 1 -- change score predicting end-of-8th-grade GPA**

<table>
<thead>
<tr>
<th>School-focused Identities</th>
<th>Model 1</th>
<th></th>
<th></th>
<th>Model 3</th>
<th></th>
<th></th>
<th>Model 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Effect Size (f²)</td>
<td>Power</td>
<td></td>
<td>Effect Size (f²)</td>
<td>Power</td>
<td></td>
<td>Effect Size (f²)</td>
<td>Power</td>
</tr>
<tr>
<td>Count</td>
<td>.037</td>
<td>.86</td>
<td></td>
<td>.039</td>
<td>.87</td>
<td></td>
<td>.039</td>
<td>.87</td>
</tr>
<tr>
<td>Balance</td>
<td>.033</td>
<td>.81</td>
<td></td>
<td>.037</td>
<td>.86</td>
<td></td>
<td>.036</td>
<td>.85</td>
</tr>
<tr>
<td>Count + Strategies</td>
<td>.035</td>
<td>.83</td>
<td></td>
<td>.042</td>
<td>.89</td>
<td></td>
<td>.041</td>
<td>.89</td>
</tr>
<tr>
<td>Plausibility</td>
<td>.051</td>
<td>.94</td>
<td></td>
<td>.030</td>
<td>.78</td>
<td></td>
<td>.028</td>
<td>.74</td>
</tr>
</tbody>
</table>

*Note: Model 1 = no controls, Model 3 = controlling for school variables and prior GPA, Model 4 = controlling for school variables, prior GPA, and demographics*

Table S5.

*Effect of interaction between school focused possible identity scores and school-level context variables on 8th grade core GPA*

<table>
<thead>
<tr>
<th>Predictor</th>
<th>School-Level FRPL</th>
<th></th>
<th>School Level Latinx</th>
<th></th>
<th>Student/Teacher Ratio</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>95% CI</td>
<td>p</td>
<td>B</td>
<td>95% CI</td>
<td>p</td>
</tr>
</tbody>
</table>


### School-focused possible identity count

<table>
<thead>
<tr>
<th>ID</th>
<th>Possible Self</th>
<th>CODE</th>
<th>Strategy</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On honor roll</td>
<td>1</td>
<td>Study every day</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>More popular</td>
<td>2</td>
<td>Hang out after school</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Playing my x-box</td>
<td>5</td>
<td>Do my chores</td>
<td>5</td>
</tr>
</tbody>
</table>

### Notes
In each model the dependent variable is 8th grade core grade point average; the regression coefficients in each cell are for the interaction between the possible identity metric in the left column and the school-level context variables in the top row.

### Section 2: File Setup and Python Code for Developing and Using Possible Selves Classifier

#### File Setup

The code is written to read possible self training data in the following format (The code ensures that only the possible self code is used to train the algorithm.)
The code is written to classify or code uncoded possible self data in the following format:

<table>
<thead>
<tr>
<th>ID</th>
<th>Possible Self</th>
<th>Strategy</th>
<th>Combined</th>
<th>Vecs</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On honor roll</td>
<td>Study every day</td>
<td>['honor', 'roll', 'study', 'every', 'day']</td>
<td>[-0.0802 - 0.34582]</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Playing my x-box</td>
<td>Do my chores</td>
<td>['Playing', 'x-box', 'chores']</td>
<td>[-0.02734 0.103027]</td>
<td>0</td>
</tr>
</tbody>
</table>

Running the code will output possible self classification in the format below. The ‘Combined” column shows which words in the responses were used by the algorithm. The ‘Vecs’ column can be ignored. The ‘Class’ column has the possible self code generated by the algorithm.

**Python Code**
# Import the necessary packages

```python
import pandas as pd
import gensim
from autocorrect import spell
import string
from nltk.corpus import stopwords
import numpy as np
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.model_selection import ShuffleSplit
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
import xlrd
import openpyxl
```

# Defining classification functions

```python
def CombineFiles(files):
    t = pd.DataFrame()
    list = []
    for file in files:
        data = pd.read_excel(file)
        list.append(data)
    return pd.concat(list, ignore_index=True)

def listCreation(X):
    X_n = []
```
for i in X:
    X_n.append(i)
return X_n

def combineClass(class_codes,t):
    df = t.copy()
    for i in class_codes:
        df.ix[df.CODE == i, 'CODE'] = 0
    return df

def classifyingResult(cv,classifier,X,Y):
    accuracies = []
    for train_index, test_index in cv.split(X):
        X_tr = [X[i - 1] for i in train_index]
        X_tes = [X[i - 1] for i in test_index]
        y_tr = [Y[i - 1] for i in train_index]
        y_tes = [Y[i - 1] for i in test_index]
        clf = classifier.fit(X_tr, y_tr)
        y_pred = clf.predict(X_tes)
        a = accuracy_score(y_tes, y_pred)
        c = confusion_matrix(y_tes, y_pred)
        accuracies.append(a)
        print a
        print c
        print "Mean Accuracy : ", np.mean(accuracies)

def classifyingResultWithSeprateTrainTest(X_tr,y_tr,X_tes):
    classifier_SVM = svm.SVC(kernel='linear',decision_function_shape='ovr')
classifier = classifier_SVM
clf = classifier.fit(X_tr, y_tr)
y_pred = clf.predict(X_tes)
t_test['Class'] = y_pred
t_test.to_excel('output_cycle.xlsx')

#Defining stopwords and importing word representations
#Google doc2vec file location can be downloaded from https://github.com/mmihaltz/word2vec-GoogleNews-vectors

stop = set(stopwords.words('english'))
model = gensim.models.KeyedVectors.load_word2vec_format('GoogleNews-vectors-negative300.bin', binary=True)

#This section first combines files with training data (if there are multiple files) then preprocess them (including ensuring there is a single code -- the possible self code -- for each response), and combines necessary classes (here, 3, 4, 5, and 7) into one class

files = [[Names of Files with training data in them]]
t = CombineFiles(files)
t = t[pd.notnull(t['CODE'])]
t.loc[t['CODE'] != t['CODE.1'], 'Strategy'] = ''
t['Combined'] = t['Possible Self'] + ' ' + t['Strategy']
t['Combined'].fillna('', inplace=True)
t['Combined'] = t['Combined'].apply(lambda x: x.encode('utf-8'))
t['Combined'] = t['Combined'].apply(lambda x: str(x))
t['Combined'] = t['Combined'].apply(lambda x: x.lower())
t['Combined'] = t['Combined'].apply(lambda x: x.translate(None, string.punctuation))
t['Combined'] = t['Combined'].apply(lambda x: [spell(str(item)) for item in x.split() if spell(str(item)) not in stop and spell(str(item)) in model.vocab]}
t['Vecs'] = t['Combined'].apply(lambda s: reduce(lambda x, y: x + y, map(lambda e: np.array(model[e]), s)) if len(s)!=0 else np.zeros(300, dtype='float32'))

original = t.copy()
class_codes= [3,4,5,7]  # combining classes 3, 4, 5, 7 (or whichever classes you wish to combine into one class)
t = combineClass(class_codes,original)
X = t['Vecs']
Y = t['CODE']
X = listCreation(X)
Y = listCreation(Y)

print confusion_matrix(Y,Y)  # To get an idea of the data distribution

# Split the data based on parameters provided
n_splits = 10
test_size = 0.15

# 10-fold cross validation SVM classification
cv = ShuffleSplit(n_splits=n_splits, test_size=test_size, random_state=0)
classifier_SVM = svm.SVC(kernel='linear', decision_function_shape='ovr')
classifier = classifier_SVM
classifyingResult(cv,classifier,X,Y)

# this section of the code should be run if a separate test file needs to be coded. First, it trains on the combined pre-processed files from the previous portion, then it preprocesses the test file, and runs the classification

files_test = [Name of file with data to be coded]
t_test = CombineFiles(files_test)
t_test['Strategy'].fillna('', inplace=True)
t_test['Possible Self'].fillna('', inplace=True)
t_test['Combined'] = t_test['Possible Self'] + ' ' + t_test['Strategy']
t_test['Combined'].fillna('', inplace=True)
t_test['Combined'] = t_test['Combined'].apply(lambda x: x.encode('utf-8'))
t_test['Combined'] = t_test['Combined'].apply(lambda x: str(x))
t_test['Combined'] = t_test['Combined'].apply(lambda x: x.lower())
t_test['Combined'] = t_test['Combined'].apply(lambda x: [spell(str(item)) for item in x.split() if spell(str(item)) not in stop and spell(str(item)) in model.vocab])
t_test['Vecs'] = t_test['Combined'].apply(lambda s: reduce(lambda x, y: x + y, map(lambda e: np.array(model[e]), s)) if len(s)!=0 else np.zeros(300, dtype='float32'))
original_test = t_test.copy()

X_test = t_test['Vecs']
X_test = listCreation(X_test)

classifyingResultWithSeprateTrainTest(X,Y,X_test)