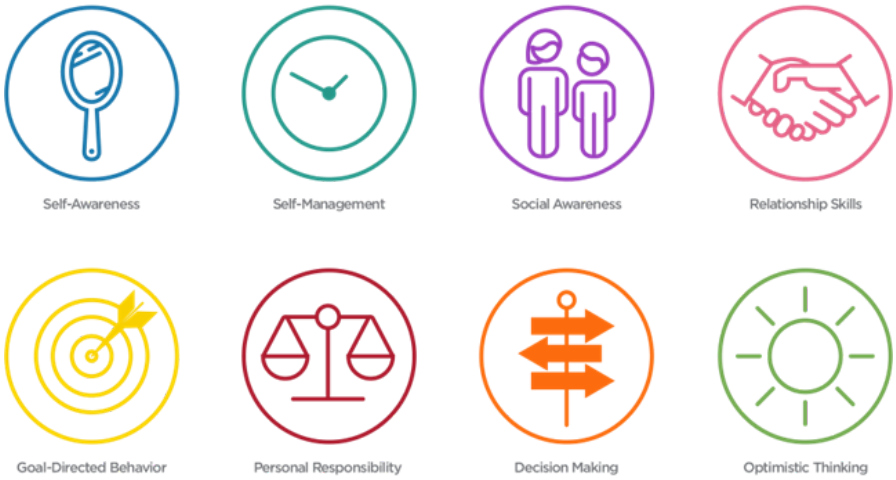
```
Course Project: Milestone 4

College of Science and Technology, Bellevue University

DSC630-T301: Predictive Analytics

Alyssa Weber

June 3, 2023
```

# Introduction

## Project Proposal

Since the COVID pandemic of 2020, there has been an observed increase of prioritization in education regarding Social Emotional Learning (SEL). The lockdowns of the pandemic isolated adolescents and this period “significantly moderated changed in depressive symptoms, anxiety, and life satisfaction” (Magson et al., 2021). I have been employed in the White Bear Lake District for eight years as a mathematics teacher. Our high school has set a building goal for the 2022-2023 school year stating that “100% of student and staff will feel their social emotional needs are being supported”. Students are given a survey called DESSA (Devereux Student Strengths Assessment) using a company called Aperture Education to track student self-reported SEL scores in eight different strands, shown below. The survey this year was administered in late November and again in late March. White Bear Lake Area High School would like to know what features, other than survey scores, could be used as earlier predictors of social emotional wellbeing.




Dessa Survey Strands (9-12 sel, 2023)

Not only is this analysis on student SEL interesting, but it is also necessary. If educators could accurately predict at-risk students early on in their high school careers, we could implement earlier forms of intervention. Increasing student emotional well-being will lead to safer learning environments, higher levels of performance, and greater social connections. The only way that students can succeed as scholars and leaders in their communities is to make sure their crucial needs are being met first. The CDC (Center for Disease Control and Prevention) reports that the levels of trauma and distress among adolescents observed in the years of 2011-2021 requires action (CDC, 2003). The sooner interventions can take place, the better.

This project would directly support the students enrolled at my high school. Administrators, support staff, and educators are presently using data reactively, rather than proactively, to target students in need of support mid-year. Data for this analysis has been put together by the director of teaching and learning who is also a part of the research, evaluation, and assessment team. The data includes grade level, gender, resolved race/ethnicity, zip code, home language, special education tags, 504 support tags, English language learner tags, total college credits taken and the latest date in which they were taken. It also includes tThis project would directly support the students enrolled at my high school. Administrators, support staff, and educators are presently using data reactively, rather than proactively, to target students in need of support mid-year. Data for this analysis has been put together by the director of teaching and learning who is also a part of the research, evaluation, and assessment team. The data includes grade level, gender, resolved race/ethnicity, zip code, home language, special education tags, 504 support tags, English language learner tags, total college credits taken and the latest date in which they were taken. It also includes the target variable of fall SEC (Social-Emotional Composite) scores. The target variable of fall SEC (Social-Emotional Composite) scores.

## Model Proposal

Regression models will work best for the original dataset since it would be predicting numerical SEL scores. However, I also plan on converting the SEL scores into categorical data such as “needs support” and “typical” with the cutoff scores aligning with the DESSA survey scales. In this case I could create a logistic regression model and/or a decision tree. These models will be run using the sklearn packages including “LinearRegression”, “LogisticRegression”, and “DecisionTreeClassifier”. I plan on using cross-validation techniques or an 80/20 training and testing split. Results for the linear regression can be evaluated using sklearn’s accuracy tool which provides an r-squared value. This will help determine how much the given variables account for the variation of the target scores and how much variation is due to other variables. Results for the logistic regression and decision tree can be found using sklearn’s accuracy score and confusion matrix which commonly uses accuracy, recall, precision, and F1 scores. I will specifically be using the confusion matrix to evaluate false positives and negatives.

## Ethical Implications

I hope to gain more insight on the social emotional well-being of the secondary students in my district. Prioritizing strong relationships with our most at-risk youth and supporting their emotional needs helps create a fostering and productive classroom culture. Ethically, this data was collected from students in our district that staff work with daily, and the results will only be able to generalize to the students within the district. Conclusions will need to be interpreted accurately and presented thoughtfully as I talk about the mental health of our adolescents. False negative results (identifying a student as “does not need support”, when in fact they do) will be more harmful than false positives. However, luckily for us, self-reported survey results will be not too far behind to continue to guide our work. As educators, we are intentional about not developing preconceptions about our students and maintaining a growth-mindset. Predicting a student’s need for emotional support through background knowledge and algorithms can take away some of our natural instincts to initiate support. Lastly, there are student privacy data laws that will need to be taken seriously in this analysis. Reporting specific cases should only involve personnel that work directly with the student. Whole group data should be collected and presented with student anonymity as a top priority.

# Methods/ Results

## Data Exploration

I began the EDA process by making sure that the dataset was large and robust enough to provide insight on SEL scores for secondary students in my district.

In [1]:
```
# used to create and work with data frames
import pandas as pd
```

In [2]:
```
# import the data and confirm that it loaded correctly
dessa_df = pd.read_csv("DESSA Data Names Removed (1).csv")
dessa_df.head()
```

Out[2]:

| | Grade | Gender | Resolved Race/Ethnicity | Home ZIP Code | Sped | 504 | EL | Home Language | GPA | Latest College Credit Taken | Total College Credit Taken | Fall SEC TScore | Fall SEC Percentile | Fall SEC Description | Winter SEC TScore | Wint SE Percent |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9 | Female | White | 55014 | N | NaN | NaN | English (011) | 1.612 | NaN | NaN | 28.0 | 1.0 | N | NaN | Na |
| **1** | 11 | Female | White | 55014 | N | NaN | NaN | English (011) | 1.867 | NaN | NaN | 36.0 | 8.0 | N | NaN | Na |
| **2** | 12 | Male | White | 55014 | N | NaN | NaN | English (011) | 2.601 | 2023.0 | 4.0 | 42.0 | 21.0 | T | NaN | Na |
| **3** | 9 | Female | White | 55014 | N | NaN | NaN | English (011) | 1.600 | NaN | NaN | 57.0 | 76.0 | T | 52.0 | 58 |
| **4** | 11 | Female | White | 55014 | N | NaN | NaN | English (011) | 1.722 | 2021.0 | 1.0 | 60.0 | 84.0 | S | 64.0 | 92 |

In [3]:
```
# view the number of rows and columns
dessa_df.shape
```

Out[3]: (2520, 17)

The data set originally started with 2,520 students. Since the model is going to be used to predict SEL scores, it is imperative that it only includes students who have a Fall SEC score.

There were 1,052 students who did not take the fall survey and were therefore removed from the dataset. It would not be beneficial to fill scores with a mean value for a target variable. It is worrisome that the remaining scores may be biased against students who chose to participate in the survey. However, with the goal of identifying at risk students, 1,468 remaining instances should still provide adequate guidance.

In [4]:
```
# view how many missing Fall SEC scores there are
dessa_df["Fall SEC TScore"].isna().sum()
```

Out[4]: 1052

In [5]:
```
# drop NaN Fall SEC scores
dessa_df = dessa_df.dropna(subset=["Fall SEC TScore"])
```

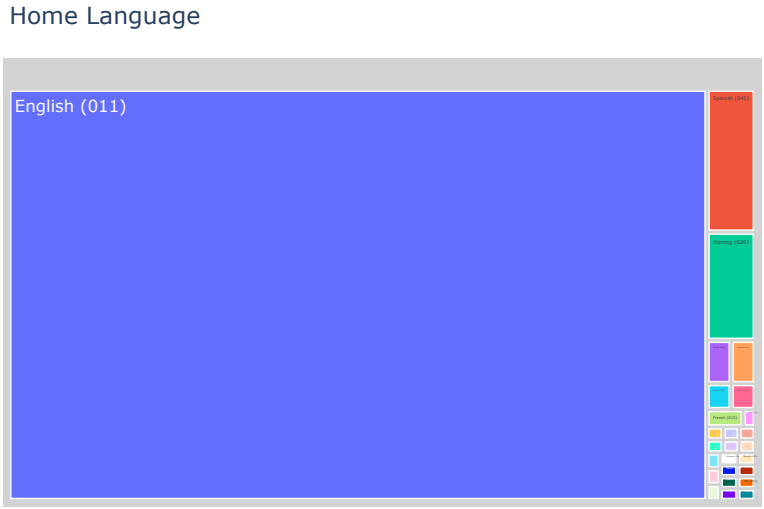
The original dataset provided 17 features. I have removed the fall SEC percentile and the winter SEC percentile as they are not useful training or testing features. There is a column named “Fall SEC Description” that categorizes the fall SEC scores into three different sets; N- “Needs Support”, T – “Typical”, and S – “Strength”. Since I am only interested in classifying students as either “Needs Support” or does not need support, I have changed all N values to 1, and both T and S values to 0. This column will be used for a classification model. The raw SEC score column will be used in a regression model. There are two columns that contain Winter SEC scores. I will keep these in the original data frame but will withhold them from now in the condensed modeling data frame.

In [6]:
```
# drop percentage columns
dessa_df = dessa_df.drop('Winter SEC Percentile', axis=1)
dessa_df = dessa_df.drop('Fall SEC Percentile', axis=1)

# Change Needs Support (N) to a value of 1
# Change Typical (T) and Strength (S) to a value of 0
dessa_df['Fall SEC Description'] = dessa_df['Fall SEC Description'].replace(['N','T', 'S'], ['1','0', '0'])
```

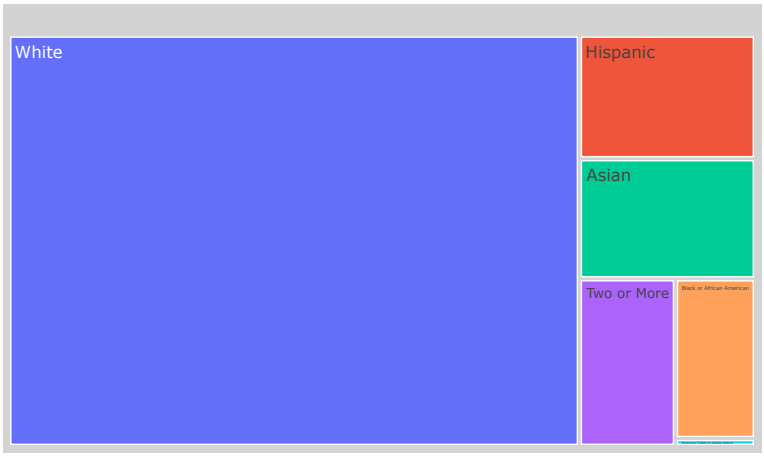
To explore the data, I used a variety of visualizations. To begin, I used tree maps for categorical features to verify that there was enough data in each subgroup to use in the modeling phase. From this exploratory process, I decided to remove the “Home Language” column. 93% of the students in this dataset use English at home. 23 other subgroups had counts of less than five, making this feature not reliable for a test/training split. I ran into errors when trying to build visualizations for 504 Services and English Language Learners. For both features I found that over 93% of the data was missing in each column and therefore would also be removed. In this process I also investigated two columns named “Latest College Credit Taken” and “Total College Credit Taken". Both columns appear to give similar information in a slightly different format. I decided to keep "Total College Credits Taken"

In [7]:
```
# for advanced area and stacked area charts
import plotly.express as px

# turns off warnings
# The tree map produces future warnings using plotly
import warnings
warnings.filterwarnings('ignore')
```

In [8]:
```
# create a treemap to view Home Language categories
fig = px.treemap(dessa_df, path=['Home Language'],
                 title="Home Language", width=600, height=400)
fig.update_traces(root_color="lightgrey")
fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
fig.show()
```




In [9]:
```
# create a treemap to view Race/Ethnicity categories
fig = px.treemap(dessa_df, path=['Resolved Race/Ethnicity'],
                 title="Resolved Race/Ethnicity", width=600, height=400)
fig.update_traces(root_color="lightgrey")
fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
fig.show()
```



In [10]:
```
# create a treemap to view Race/Ethnicity categories
fig = px.treemap(dessa_df, path=['Grade'],
                 title="Grade", width=600, height=400)
fig.update_traces(root_color="lightgrey")
fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
fig.show()
```


In [11]:
```
# create a treemap to view Race/Ethnicity categories
fig = px.treemap(dessa_df, path=['Sped'],
                 title="Sped", width=600, height=400)
fig.update_traces(root_color="lightgrey")
fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
fig.show()
```


In [12]:
```
#produces errors

# create a treemap to view 504 categories
#fig = px.treemap(dessa_df, path=['504'],
                 #title="504")
#fig.update_traces(root_color="lightgrey")
#fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
#fig.show()

# number of NaN values in 504
dessa_df["504"].isna().sum()
```

Out[12]: 1395

In [13]:
```
#produces errors

# create a treemap to view EL categories
#fig = px.treemap(dessa_df, path=['EL'],
                 #title="EL")
#fig.update_traces(root_color="lightgrey")
#fig.update_layout(margin = dict(t=50, l=25, r=25, b=25))
#fig.show()

# number of NaN values in EL
dessa_df["EL"].isna().sum()
```

Out[13]: 1377

In [14]:
```
# number of NAN values in Total College Credits Taken
dessa_df["Total College Credit Taken"].isna().sum()
```

Out[14]: 864

In [15]:
```
# number of NAN values in Latest College Credits Taken
dessa_df["Latest College Credit Taken"].isna().sum()
```

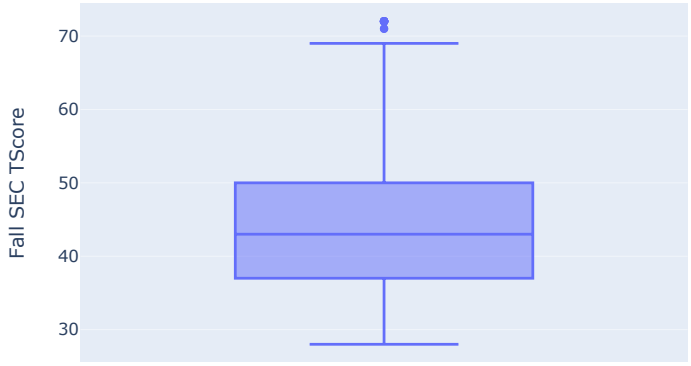
Out[15]: 864

In [16]:
```
dessa_df = dessa_df.drop('Home Language', axis=1)
dessa_df = dessa_df.drop('504', axis=1)
dessa_df = dessa_df.drop('EL', axis=1)
dessa_df = dessa_df.drop('Latest College Credit Taken', axis=1)
```

To view the spread of numerical data I chose to use boxplots. For GPA I found that all of the scores are between 0-4.5. 75% of the students had a GPA in the range of 2.5-4.5 with even spread. The first 25% of the GPA data was spread out between 0-2.5 with quite a few outliers. These outliers will be kept without change in the dataset due to accurate representations. I also viewed the spread of the Fall SEC scores. I found that my target variable is slightly skewed right with two upper outliers. It is likely that these students took the survey and answered every question with the highest rank. It is hard to determine whether these are accurate reflections of the students' self-evaluation or not. These data points will also be kept in the dataset.

In [17]:
```
# spread of GPA data
fig = px.box(dessa_df, y="GPA", width=600, height=400)
fig.show()
```

In [18]:
```
# spread of Fall SEC scores
fig = px.box(dessa_df, y="Fall SEC TScore", width=600, height=400)
fig.show()
```




My final steps for cleaning the data were to fill NaN values in the “Total College Credit Taken” column with “0” since these students have not completed a college credit and to create dummy features for categorical data. I created two condensed data frames named “dessa_df_dummy” and “dessa_df_dummy_lin”. The original data frame was used for the categorical descriptions of Needs Support vs Does Not Need Support, while the latter was used for the linear regression with the original SEC scores. The final data frame to be used in the modeling phase consists of:

1. Categorical dummy features including:

- Grade
- Gender
- Resolved Race/Ethnicity
- Home ZIP Code
- Sped

2. Numerical features including:

- GPA
- Total College Credit Taken

3. Target Variable:

- Fall SEC Score (Numerical)
- Fall SEC Description (Categorical)

In [19]:
```
# Create smaller data frame with by dropping winter scores and numerical fall scores
condensed_df = dessa_df.drop('Winter SEC TScore', axis=1)
condensed_df = condensed_df.drop('Winter SEC Description', axis=1)
condensed_df = condensed_df.drop('Fall SEC TScore', axis=1)
condensed_df = condensed_df.fillna(0)
#condensed_df.head(5)
```

In [20]:
```
# create dummy variables for categorical data
dessa_df_dummy = pd.get_dummies(condensed_df, columns=['Grade', 'Total College Credit Taken',
                                                       'Gender', 'Resolved Race/Ethnicity', 'Sped',
                                                       'Home ZIP Code'])
#dessa_df_dummy.head()
```

In [21]:
```
# Create smaller data frame with by dropping winter scores and fall description scores
condensed_df = dessa_df.drop('Winter SEC TScore', axis=1)
condensed_df = condensed_df.drop('Winter SEC Description', axis=1)
condensed_df = condensed_df.drop('Fall SEC Description', axis=1)
condensed_df = condensed_df.fillna(0)
#condensed_df.head(5)
```

In [22]:
```
# create dummy variables for categorical data
dessa_df_dummy_lin = pd.get_dummies(condensed_df, columns=['Grade', 'Total College Credit Taken',
                                                           'Gender', 'Resolved Race/Ethnicity', 'Sped',
                                                           'Home ZIP Code'])
#dessa_df_dummy_lin.head()
```

## Logistic Regression Model

To perform the logistic model, I first split the data into an 80% training and 20% testing set with the Fall SEC Description as the target feature. I then created the Logistic Regression Model using the sklearn package for both the modeling and the classification reports. The Logistic Regression Model produced an accuracy score of 63% on the holdout data. To analyze if these results were meaningful or not, I produced both a confusion matrix and ran a classification report.

In [23]:
```
# split the data
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(dessa_df_dummy.drop('Fall SEC Description',axis=1),
                                                    dessa_df_dummy['Fall SEC Description'], test_size=0.20,
                                                    random_state=101)
```

In [24]:
```
# create, fit and make predictions from a logistic regression model
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
predictions = logmodel.predict(X_test)
```

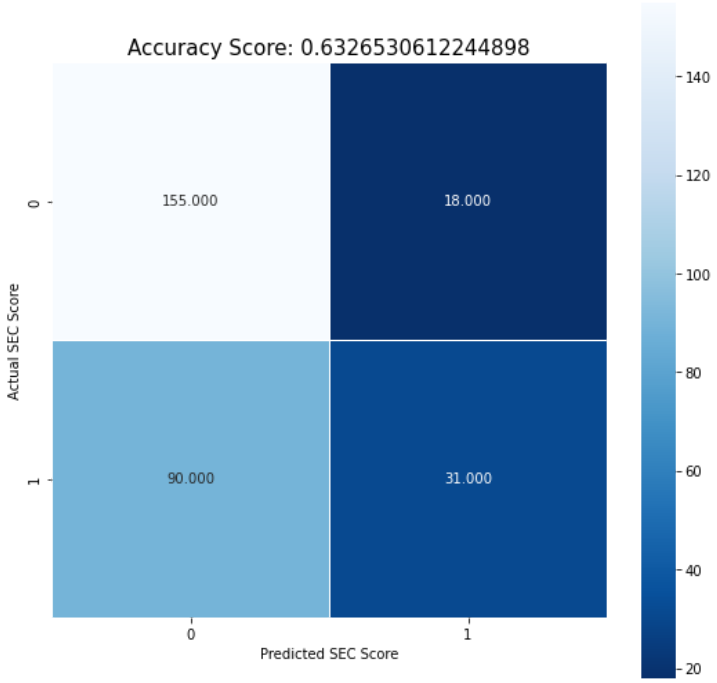
In [25]:
```
# calculate the accuracy score
from sklearn.metrics import accuracy_score
score = accuracy_score(y_test, predictions)
score
```

Out[25]:
```
0.6326530612244898
```

The accuracy of the logistic regression model was 63%. Meaning that the model predicted the SEL description as "Needs Support" or "Does Not Need Support" 63% of the time. This model only produced 18 false positives, indicating the model thought a student would need support, when in fact they did not. Many of the inaccurate results were false negatives, implying the model predicted that students would not need support, when in fact they did.

In [26]:
```
from sklearn import metrics
from matplotlib import pyplot as plt # visualizations
import seaborn as sns # additional visualizations
```

In [27]:
```
# create a confusion matrix
cm = metrics.confusion_matrix(y_test, predictions)
plt.figure(figsize=(9,9))
sns.heatmap(cm, annot=True, fmt=".3f", linewidths=.5, square = True, cmap = 'Blues_r');
plt.ylabel('Actual SEC Score');
plt.xlabel('Predicted SEC Score');
all_sample_title = 'Accuracy Score: {0}'.format(score)
plt.title(all_sample_title, size = 15)
plt.show()
```




In the table below, we can see that the F-1 score for Does Not Need Support (0) is 0.74, while the score for Needs Support (1) is 0.36. This shows us that this model works best for classifying students who do not need support.

In [28]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.63      0.90      0.74       173
           1       0.63      0.26      0.36       121

    accuracy                           0.63       294
   macro avg       0.63      0.58      0.55       294
weighted avg       0.63      0.63      0.59       294
```

## Decision Tree Model

I built a decision tree next with sklearn's DecisionTreeClassifier package with the hopes of raising accuracy and dropping the false negative rate. I used a new 80/20% training testing split. This model gave me a decreased accuracy score of 56%.

In [29]:
```
# load decision tree classifier library
from sklearn.tree import DecisionTreeClassifier
```

In [30]:
```
# split the data
X_train, X_test, y_train, y_test = train_test_split(dessa_df_dummy.drop('Fall SEC Description',axis=1),
                                                    dessa_df_dummy['Fall SEC Description'], test_size=0.20,
                                                    random_state=35)
```

In [31]:
```
# create a decision tree classifier object
# The random_state of 10 (an integer) creates replicable results
decisiontree = DecisionTreeClassifier(random_state=10)
```

In [32]:
```
# train the model
tree_model = decisiontree.fit(X_train, y_train)
```

In [33]:
```
# predicting the test set results
y_pred = tree_model.predict(X_test)
```

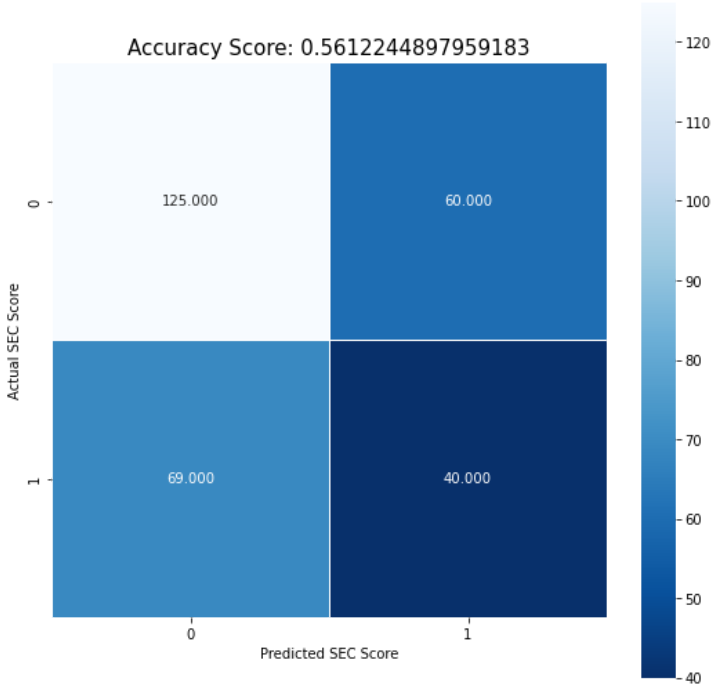
In [34]:
```
# calculate the accuracy score
from sklearn.metrics import accuracy_score
score = accuracy_score(y_test, y_pred)
score
```

Out[34]: 0.5612244897959183

The confusion matrix below shows more false positives than the previous model and less false negatives. Although this seems good at first glance, it is troublesome for the White Bear Lake District in trying to make predictions. This model would end up targeting 100 students to receive interventions, when only 40 of them needed them. The previous model only targeted 49 students, but 31 students did in fact need intervention. The logistic model would be a better allocation of resources and funding as the goal is to accurately target a subgroup of students who could use early intervention.

In [35]:
```
# create a confusion matrix
cm = metrics.confusion_matrix(y_test, y_pred)
```

In [36]:
```
# view the confusion matrix
plt.figure(figsize=(9,9))
sns.heatmap(cm, annot=True, fmt=".3f", linewidths=.5, square = True, cmap = 'Blues_r');
plt.ylabel('Actual SEC Score');
plt.xlabel('Predicted SEC Score');
all_sample_title = 'Accuracy Score: {0}'.format(score)
plt.title(all_sample_title, size = 15)
plt.show()
```




The classification report for the Decision Tree shows that the F-1 scores again favor this model for predicting students who do not need support (0). Since the accuracy went down and the false positives rose dramatically, this model will not be presented to the district.

In [37]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.64      0.68      0.66       185
           1       0.40      0.37      0.38       109

    accuracy                           0.56       294
   macro avg       0.52      0.52      0.52       294
weighted avg       0.55      0.56      0.56       294
```

## Linear Regression Model

I also ran a Linear Regression Model using a new 80/20% training/test split on the data frame containing the raw SEC scores. All the variables listed in the data exploration section were used to predict the SEC numerical score. To measure its performance, I started with the r^2 score which reported at a measly 0.11. This quantity, being closer to 0 than 1, suggests that this linear model only explains 11% of the variation in SEC scores. I did not pursue this model further after this result. The district is not as concerned anyways with the exact score of each student, but rather if the student needs support or not.

In [38]:
```
# split the data
X_train, X_test, y_train, y_test = train_test_split(dessa_df_dummy_lin.drop('Fall SEC TScore',axis=1),
                                                    dessa_df_dummy_lin['Fall SEC TScore'], test_size=0.20,
                                                    random_state=12)
```

In [39]:
```
# create a linear model from the training data
from sklearn.linear_model import LinearRegression
model_fit = LinearRegression()
model_fit.fit(X_train, y_train)
```

Out[39]:
```
LinearRegression()
```

In [40]:
```
# predicting the training set results
y_pred_train = model_fit.predict(X_train)

# predicting the test set results
y_pred = model_fit.predict(X_test)
```

In [41]:
```
#calculate R2 on the training dataset
from sklearn.metrics import r2_score
r2_score(y_train, y_pred_train)
```

Out[41]:
```
0.1186722789657868
```

# Conclusion

Students are experiencing higher levels of social-emotional distress which is impacting their education. In this study of White Bear Lake Area secondary students, I have explored three models to determine the best one for predicting students SEC scores or levels in the hopes of deploying a model that could help our district identify at risk students earlier within the school year. I have learned that this is not a task to be taken lightly, and that the details of the results impact the decision-making process the most. Based off my analysis, I would suggest that the district deploy the logistic model at the start of the year before the surveys are taken. All students predicted to need support should be given early intervention opportunities.

Ethically, this model inherently has false positive and negative results that need to be taken into consideration. Survey biases need to be kept at the forefront of decision making when it comes to self-reported mental health states. Luckily for the district, it is only trying to increase early interventions, rather than reallocate resources involving this matter. It is likely that <10% of the students will be labeled as needing support, when in fact they do not. It will not be harmful for students who do not need social-emotional intervention to be given intervention tasks that help build overall well-being. It is a priority for the district, however, that the students correctly categorized begin the process as soon as possible. Based on the model results of this year's data, we will miss many students who still need support.

My further recommendation to the district is to follow up with the Dessa Survey scores in November to target the rest of the students that need intervention. I would like to continue to monitor and adjust the model as we collect further years of data.