

**Rubric for Module 18 Challenge:**

	<b>Proficiency 30 to &gt; 27 points</b>	<b>Approaching Proficiency 27 to &gt; 24 points</b>	<b>Developing Proficiency 24 to &gt; 20 points</b>	<b>Emerging 20 to &gt; 0 points</b>	<b>Incomplete</b>
<b>Deliverable 1: Preprocessing the Data for PCA</b>	<ul style="list-style-type: none"> <li>The following preprocessing steps have been performed on the crypto_df DataFrame: (16 pt)               <ul style="list-style-type: none"> <li>All cryptocurrencies that are not being traded are removed</li> <li>All cryptocurrencies with at least one null value are dropped</li> <li>All the rows that do not have coins being mined are removed</li> <li>The IsTrading column is dropped</li> <li>The CoinName column is dropped from the crypto_df DataFrame</li> </ul> </li> </ul> <p>AND all the following have been completed:</p> <ul style="list-style-type: none"> <li>A new DataFrame is created that stores the names of all cryptocurrencies from the CoinName column and has the index from the crypto_df DataFrame (4 pt)</li> <li>The get_dummies() method is used to create variables for all of the text features, which are then stored in a new DataFrame, X (5 pt)</li> <li>The features from the X DataFrame have been standardized using the StandardScaler fit_transform() function (5 pt)</li> </ul>	<ul style="list-style-type: none"> <li>The following preprocessing steps have been performed on the crypto_df DataFrame: (16 pt)               <ul style="list-style-type: none"> <li>All cryptocurrencies that are not being traded are removed</li> <li>All cryptocurrencies with at least one null value are dropped</li> <li>All the rows that do not have coins being mined are removed</li> <li>The IsTrading column is dropped</li> <li>The CoinName column is dropped from the crypto_df DataFrame</li> </ul> </li> </ul> <p>AND all the following have been completed:</p> <ul style="list-style-type: none"> <li>A new DataFrame is created that stores the names of all cryptocurrencies with the CoinName column and has the index from the crypto_df DataFrame (4 pt)</li> <li>The get_dummies() method is used to create variables for the TWO text features, which are then stored in a new DataFrame, X (5 pt)</li> <li>The features from the X DataFrame have been standardized using the StandardScaler fit_transform() function, but there is an error (2 pt)</li> </ul>	<ul style="list-style-type: none"> <li>The following preprocessing steps have been performed on the crypto_df DataFrame: (14 pt)               <ul style="list-style-type: none"> <li>All cryptocurrencies that are not being traded are removed</li> <li>All cryptocurrencies with at least one null value are dropped</li> <li>All the rows that do not have coins being mined are removed</li> <li>The IsTrading column is dropped</li> </ul> </li> </ul> <p>AND all the following have been completed:</p> <ul style="list-style-type: none"> <li>A new DataFrame is created that stores the names of all cryptocurrencies with the CoinName column and has the index from the crypto_df DataFrame (4 pt)</li> <li>The get_dummies() method is used to create variables for the ONE of the TWO text features, which are then stored in a new DataFrame, X (3 pt)</li> <li>The features from the X DataFrame have been standardized using the StandardScaler fit_transform() function (3 pt)</li> </ul>	<ul style="list-style-type: none"> <li>The following preprocessing steps have been performed on the crypto_df DataFrame: (12 pt)               <ul style="list-style-type: none"> <li>All cryptocurrencies that are not being traded are removed</li> <li>All cryptocurrencies with at least one null value are dropped</li> <li>All the rows that do not have coins being mined are removed</li> </ul> </li> </ul> <p>AND all the following have been completed:</p> <ul style="list-style-type: none"> <li>A new DataFrame is created that stores the names of all cryptocurrencies with the CoinName column BUT does not have the index from the crypto_df DataFrame (2 pt)</li> <li>The get_dummies() method is used to create variables for the ONE of the TWO text features, which are then stored in a new DataFrame, X (3 pt)</li> <li>The features from the X DataFrame have been standardized using the StandardScaler fit_transform() function (3 pt)</li> </ul>	<p>No submission was received</p> <p>-OR-</p> <p>Submission was empty or blank</p> <p>-OR-</p> <p>Submission contains evidence of academic dishonesty</p>
	<b>Proficiency</b>	<b>Approaching Proficiency</b>	<b>Developing Proficiency</b>	<b>Emerging</b>	

	20 to > 18 points	18 to > 16 points	16 to > 14 points	14 to > 0 points	
<b>Deliverable 2: Reducing Data Dimensions Using PCA</b>	<ul style="list-style-type: none"> <li>The PCA algorithm reduces the dimensions of the X DataFrame down to three principal components. (10 pt)</li> <li>The pcs_df DataFrame is created (5 pt)</li> </ul> <p>And, all the following are completed:</p> <ul style="list-style-type: none"> <li>The pcs_df DataFrame has three columns; PC 1, PC 2, and PC 3 (2 pt)</li> <li>The pcs_df DataFrame uses the index from the crypto_df DataFrame (3 pt)</li> </ul>	<ul style="list-style-type: none"> <li>The PCA algorithm reduces the dimensions of the X DataFrame down to three principal components. (10 pt)</li> <li>✓ The pcs_df DataFrame is created (5 pt)</li> </ul> <p>And, ONE of the following are completed:</p> <ul style="list-style-type: none"> <li>The pcs_df DataFrame has three columns; PC 1, PC 2, and PC 3 (2 pt)</li> <li>The pcs_df DataFrame uses the index from the crypto_df DataFrame (3 pt)</li> </ul>	<ul style="list-style-type: none"> <li>The PCA algorithm reduces the dimensions of the X DataFrame down to three principal components. (10 pt)</li> <li>The pcs_df DataFrame is created (5 pt)</li> <li>Code is written to add the three columns; PC 1, PC 2, and PC 3 to the pcs_df DataFrame (1 pt)</li> </ul>	<ul style="list-style-type: none"> <li>The PCA algorithm reduces the dimensions of the X DataFrame down to three principal components. (10 pt)</li> <li>Code is written to create the pcs_df DataFrame with the three columns; PC 1, PC 2, and PC 3 (4 pt)</li> </ul>	
	<b>Proficiency 20 to &gt; 18 points</b>	<b>Approaching Proficiency 18 to &gt; 15 points</b>	<b>Developing Proficiency 15 to &gt; 12 points</b>	<b>Emerging 12 to &gt; 0 points</b>	
<b>Deliverable 3: Clustering Cryptocurrencies Using K-means</b>	<ul style="list-style-type: none"> <li>An Elbow Curve is created using hvPlot to find the best value for K (10 pt)</li> <li>Predictions are made on the K clusters of the cryptocurrencies' data (5 pt)</li> <li>A new DataFrame is created with the same index as the crypto_df DataFrame and has NINE columns (5 pt)</li> </ul>	<ul style="list-style-type: none"> <li>An Elbow Curve is created using hvPlot to find the best value for K (10 pt)</li> <li>Predictions are made on the K clusters of the cryptocurrencies' data (5 pt)</li> <li>A new DataFrame is created with the same index as the crypto_df DataFrame and has SEVEN of the NINE columns (3 pt)</li> </ul>	<ul style="list-style-type: none"> <li>An Elbow Curve is created using hvPlot to find the best value for K (10 pt)</li> <li>Predictions are made on the K clusters of the cryptocurrencies' data (5 pt)</li> </ul>	<ul style="list-style-type: none"> <li>An Elbow Curve is created using hvPlot to find the best value for K (10 pt)</li> <li>Code is written to make the predictions on the K clusters of the cryptocurrencies' data (2 pt)</li> </ul>	
	<b>Proficiency 30 to &gt; 27 points</b>	<b>Approaching Proficiency 27 to &gt; 25 points</b>	<b>Developing Proficiency 25 to &gt; 22 points</b>	<b>Emerging 22 to &gt; 0 points</b>	
<b>Deliverable 4: Visualizing Cryptocurrencies Results</b>	<ul style="list-style-type: none"> <li>The clusters are plotted using a 3D-Scatter and each data point shows the CoinName and Algorithm on hover (10 pt)</li> <li>A table with tradable cryptocurrencies is created using the hvplot.table()</li> </ul>	<ul style="list-style-type: none"> <li>The clusters are plotted using a 3D-Scatter and each data point shows the CoinName and Algorithm on hover (10 pt)</li> <li>A table with tradable cryptocurrencies is created</li> </ul>	<ul style="list-style-type: none"> <li>The clusters are plotted using a 3D-Scatter and each data point shows the CoinName and Algorithm on hover (10 pt)</li> <li>A table with tradable cryptocurrencies is created</li> </ul>	<ul style="list-style-type: none"> <li>The clusters are plotted using a 3D-Scatter and each data point shows the CoinName and Algorithm on hover (10 pt)</li> <li>A table with tradable cryptocurrencies is created using the hvplot.table() function</li> </ul>	

	<p>function (3 pt)</p> <ul style="list-style-type: none"> <li>• The total number of tradable cryptocurrencies is printed (2 pt)</li> <li>• A DataFrame is created that contains the clustered_df DataFrame index, the scaled data, and the "CoinName" and "Class" columns (5 pt)</li> <li>• A hvplot scatter plot is created where the X-axis is "TotalCoinsMined", the Y-axis is "TotalCoinSupply", and the data is ordered by "Class", and when you hover over the the data it shows the "CoinName" (10 pt)</li> </ul>	<p>using the hvplot.table() function (3 pt)</p> <ul style="list-style-type: none"> <li>• The total number of tradable cryptocurrencies is printed (2 pt)</li> <li>• A DataFrame is created that contains the clustered_df DataFrame index, the scaled data, and the "CoinName" and "Class" columns (5 pt)</li> <li>• A hvplot scatter plot is created where the X-axis is "TotalCoinsMined", the Y-axis is "TotalCoinSupply", and the data is ordered by "Class", but the "CoinName" doesn't appear on hover (7 pt)</li> </ul>	<p>using the hvplot.table() function (3 pt)</p> <ul style="list-style-type: none"> <li>• The total number of tradable cryptocurrencies is printed (2 pt)</li> <li>• A DataFrame is created that contains the scaled data, the clustered_df DataFrame index, and the "CoinName" column (3 pt)</li> <li>• A hvplot scatter plot is created where the X-axis is "TotalCoinsMined", the Y-axis is "TotalCoinSupply", and when you hover over the the data it shows the "CoinName" (7 pt)</li> </ul>	<p>(3 pt)</p> <ul style="list-style-type: none"> <li>• The total number of tradable cryptocurrencies is printed (2 pt)</li> <li>• A DataFrame is created that contains the scaled data and the clustered_df DataFrame index. (2 pt)</li> <li>• A hvplot scatter plot is created where the X-axis is "TotalCoinsMined", the Y-axis is "TotalCoinSupply" (5 pt)</li> </ul>	
--	--	--	---	--	--