	Mastery	Approaching Mastery	Progressing	Emerging	Incomplete
Data Pre-Processing (30 points)	Data is appropriately pre-processed, including all of the following steps. ✓ Remove all cryptocurrencies that are not on trading. ✓ Remove all cryptocurrencies that have not an algorithm defined. ✓ Remove the IsTrading column. ✓ Remove all cryptocurrencies with at least one null value. ✓ Remove all cryptocurrencies without coins mined. ✓ Store the names of all cryptocurrencies on a DataFramed named coins_name, ✓ Use the crypto_df.index as the index for this new DataFrame. ✓ Remove the CoinName column. ✓ Create dummies variables for all the text features, store the resulting data on a DataFrame named X. ✓ Use the StandardScaler from sklearn to standardize all the data of the X DataFrame.	Data is appropriately pre-processed, including 7-9 of the following steps. ✓ Remove all cryptocurrencies that are not on trading. ✓ Remove all cryptocurrencies that have not an algorithm defined. ✓ Remove the IsTrading column. ✓ Remove all cryptocurrencies with at least one null value. ✓ Remove all cryptocurrencies without coins mined. ✓ Store the names of all cryptocurrencies on a DataFramed named coins_name, ✓ Use the crypto_df.index as the index for this new DataFrame. ✓ Remove the CoinName column. ✓ Create dummies variables for all the text features, store the resulting data on a DataFrame named X. ✓ Use the StandardScaler from sklearn to standardize all the data of the X DataFrame.	Data is pre-processed, including 4-6 of the following steps. ✓ Remove all cryptocurrencies that are not on trading. ✓ Remove all cryptocurrencies that have not an algorithm defined. ✓ Remove the IsTrading column. ✓ Remove all cryptocurrencies with at least one null value. ✓ Remove all cryptocurrencies without coins mined. ✓ Store the names of all cryptocurrencies on a DataFramed named coins_name, ✓ Use the crypto_df.index as the index for this new DataFrame. ✓ Remove the CoinName column. ✓ Create dummies variables for all the text features, store the resulting data on a DataFrame named X. ✓ Use the StandardScaler from sklearn to standardize all the data of the X DataFrame.	Data is incompletely pre-processed, including 1-3 of the following steps. ✓ Remove all cryptocurrencies that are not on trading. ✓ Remove all cryptocurrencies that have not an algorithm defined. ✓ Remove the IsTrading column. ✓ Remove all cryptocurrencies with at least one null value. ✓ Remove all cryptocurrencies without coins mined. ✓ Store the names of all cryptocurrencies on a DataFramed named coins_name, ✓ Use the crypto_df.index as the index for this new DataFrame. ✓ Remove the CoinName column. ✓ Create dummies variables for all the text features, store the resulting data on a DataFrame named X. ✓ Use the StandardScaler from sklearn to standardize all the data of the X DataFrame.	No submission was received -OR- Submission was empty or blank -OR- Submission contains evidence of academic dishonesty
Reducing Data Dimensions	✓ Use the PCA algorithm from sklearn to reduce the dimensions of the X DataFrame down to	✓ Use the PCA algorithm from sklearn to reduce the dimensions of the X DataFrame down to three	✓ Use the PCA algorithm from sklearn to reduce the dimensions of the X DataFrame down to three	✓ Use the PCA algorithm from sklearn to reduce the dimensions of the X DataFrame down to three	

Using PCA (20 points)	three principal components. ✓ Create a DataFrame named pcs_df ✓ Create three columns, called "PC 1," "PC 2," and "PC 3" ✓ Use the crypto_df.index as the index for this new DataFrame.	principal com And two of the three below steps: ✓ Create a DataFrame named pcs_df ✓ Create three columns, called "PC 1," "PC 2," and "PC 3"ponents. ✓ Use the crypto_df.index as the index for this new DataFrame.	principal components. And one of the three below steps: ✓ Create a DataFrame named pcs_df ✓ Create three columns, called "PC 1," "PC 2," and "PC 3"ponents. ✓ Use the crypto_df.index as the index for this new DataFrame.	principal components.	
Clustering Cryptocurrencies Using K-Means (20 points)	KMeans algorithm from sklearn is used to cluster the cryptocurrencies using the PCA data, including the below steps. ✓Create an Elbow Curve to find the best value for k, use the pcs_df DataFrame. ✓Predict the k clusters for the cryptocurrencies data. Use the pcs_df to run the KMeans algorithm. ✓Create a new DataFrame named clustered_df, that includes the following columns: Algorithm, ProofType, TotalCoinsMined, TotalCoinSupply, PC 1, PC 2, PC 3, CoinName, and Class. ✓ Maintains the index of the crypto_df DataFrames	KMeans algorithm from sklearn is used to cluster the cryptocurrencies using the PCA data, including three of the below steps. ✓Create an Elbow Curve to find the best value for k, use the pcs_df DataFrame. ✓Predict the k clusters for the cryptocurrencies data. Use the pcs_df to run the KMeans algorithm. ✓Create a new DataFrame named clustered_df, that includes the following columns: Algorithm, ProofType, TotalCoinsMined, TotalCoinSupply, PC 1, PC 2, PC 3, CoinName, and Class. ✓ Maintains the index of the crypto_df DataFrames	KMeans algorithm from sklearn is used to cluster the cryptocurrencies using the PCA data, including two of the below steps. ✓Create an Elbow Curve to find the best value for k, use the pcs_df DataFrame. ✓Predict the k clusters for the cryptocurrencies data. Use the pcs_df to run the KMeans algorithm. ✓Create a new DataFrame named clustered_df, that includes the following columns: Algorithm, ProofType, TotalCoinsMined, TotalCoinSupply, PC 1, PC 2, PC 3, CoinName, and Class. ✓ Maintains the index of the crypto_df DataFrames	KMeans algorithm from sklearn is used to cluster the cryptocurrencies using the PCA data, including one of the below steps. ✓Create an Elbow Curve to find the best value for k, use the pcs_df DataFrame. ✓Predict the k clusters for the cryptocurrencies data. Use the pcs_df to run the KMeans algorithm. ✓Create a new DataFrame named clustered_df, that includes the following columns: Algorithm, ProofType, TotalCoinsMined, TotalCoinSupply, PC 1, PC 2, PC 3, CoinName, and Class. ✓ Maintains the index of the crypto_df DataFrames	
Visualizing Results (30 points)	Visualizations include two scatter plots and a data table, described below, with no errors. **Scatter Plot 1** ✓Create a 3D-Scatter using Plotly Express to plot the clusters using the clustered_df DataFrame.	Visualizations include two scatter plots and a data table, described below, with some minor errors. **Scatter Plot 1** ✓Create a 3D-Scatter using Plotly Express to plot the clusters using the clustered_df DataFrame. ✓3D-Scatter plot includes the	Visualizations include two of the three visualizations described below. **Scatter Plot 1** ✓Create a 3D-Scatter using Plotly Express to plot the clusters using the clustered_df DataFrame. ✓3D-Scatter plot includes the	Visualizations include one of the three visualizations described below. **Scatter Plot 1** ✓Create a 3D-Scatter using Plotly Express to plot the clusters using the clustered_df DataFrame. ✓3D-Scatter plot includes the following parameters:	

	✓3D-Scatter plot includes the following parameters: hover_name="CoinName" and hover_data=["Algorithm"] to show this additional info on each data point. **Data Table** ✓Use hvplot.table to create a data table with all the current tradable cryptocurrencies. ✓The table should have the following columns: CoinName, Algorithm, ProofType, TotalCoinSupply, TotalCoinsMined, and Class. **Scatter Plot 2** ✓Create a scatter plot using hvplot.scatter, to present the clustered data about cryptocurrencies having x="TotalCoinsMined" and y="TotalCoinSupply" to contrast the number of available coins versus the total number of mined coins. ✓Use the hover_cols=["CoinName"] parameter to include the cryptocurrency name on each data point.	following parameters: hover_name="CoinName" and hover_data=["Algorithm"] to show this additional info on each data point. **Data Table** ✓Use hvplot.table to create a data table with all the current tradable cryptocurrencies. ✓The table should have the following columns: CoinName, Algorithm, ProofType, TotalCoinSupply, TotalCoinsMined, and Class. **Scatter Plot 2** ✓Create a scatter plot using hvplot.scatter, to present the clustered data about cryptocurrencies having x="TotalCoinsMined" and y="TotalCoinSupply" to contrast the number of available coins versus the total number of mined coins. ✓Use the hover_cols=["CoinName"] parameter to include the cryptocurrency name on each data point.	following parameters: hover_name="CoinName" and hover_data=["Algorithm"] to show this additional info on each data point. **Data Table** ✓Use hvplot.table to create a data table with all the current tradable cryptocurrencies. ✓The table should have the following columns: CoinName, Algorithm, ProofType, TotalCoinSupply, TotalCoinsMined, and Class. **Scatter Plot 2** ✓Create a scatter plot using hvplot.scatter, to present the clustered data about cryptocurrencies having x="TotalCoinsMined" and y="TotalCoinSupply" to contrast the number of available coins versus the total number of mined coins. ✓Use the hover_cols=["CoinName"] parameter to include the cryptocurrency name on each data point.	hover_name="CoinName" and hover_data=["Algorithm"] to show this additional info on each data point. **Data Table** ✓Use hvplot.table to create a data table with all the current tradable cryptocurrencies. ✓The table should have the following columns: CoinName, Algorithm, ProofType, TotalCoinSupply, TotalCoinsMined, and Class. **Scatter Plot 2** ✓Create a scatter plot using hvplot.scatter, to present the clustered data about cryptocurrencies having x="TotalCoinsMined" and y="TotalCoinSupply" to contrast the number of available coins versus the total number of mined coins. ✓Use the hover_cols=["CoinName"] parameter to include the cryptocurrency name on each data point.	