Exploratory Data Analysis Assignments

Contents

Source of Dataset:

https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/data

About Dataset:

The dataset comprises 18,000 postero-anterior (PA) CXR scans in DICOM format, which were de-identified to protect patient privacy. All images were labeled by a panel of experienced radiologists for the presence of 14 critical radiographic findings as listed below:

```
0 - Aortic enlargement
1 - Atelectasis
2 - Calcification
3 - Cardiomegaly
4 - Consolidation
5 - ILD
6 - Infiltration
7 - Lung Opacity
```

```
8 - Nodule/Mass
9 - Other lesion
10 - Pleural effusion
11 - Pleural thickening
12 - Pneumothorax
13 - Pulmonary fibrosis
```

The "No finding" observation (`14`) was intended to capture the absence of all findings above.

Columns:

- `image_id` - unique image identifier
- `class_name` - the name of the class of detected object (or "No finding")
- `class_id` - the ID of the class of detected object
- `rad_id` - the ID of the radiologist that made the observation
- `x_min` - minimum X coordinate of the object's bounding box
- `y_min` - minimum Y coordinate of the object's bounding box
- `x_max` - maximum X coordinate of the object's bounding box
- `y_max` - maximum Y coordinate of the object's bounding box

Sample of Dataset:

In [2]:
```
df = pd.read_csv('x_rays.csv')
df
```

Out[2]:

	image_id	class_name	class_id	rad_id	x_min	y_min	x_max	y_max
0	50a418190bc3fb1ef1633bf9678929b3	No finding	14	R11	NaN	NaN	NaN	NaN
1	21a10246a5ec7af151081d0cd6d65dc9	No finding	14	R7	NaN	NaN	NaN	NaN
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	R10	691.0	1375.0	1653.0	1831.0
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargement	0	R10	1264.0	743.0	1611.0	1019.0
4	063319de25ce7edb9b1c6b8881290140	No finding	14	R10	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
67909	936fd5cff1c058d39817a08f58b72cae	No finding	14	R1	NaN	NaN	NaN	NaN
67910	ca7e72954550eeb610fe22bf0244b7fa	No finding	14	R1	NaN	NaN	NaN	NaN
67911	aa17d5312a0fb4a2939436abca7f9579	No finding	14	R8	NaN	NaN	NaN	NaN
67912	4b56bc6d22b192f075f13231419dfcc8	Cardiomegaly	3	R8	771.0	979.0	1680.0	1311.0
67913	5e272e3adbdaafb07a7e84a9e62b1a4c	No finding	14	R16	NaN	NaN	NaN	NaN

67914 rows × 8 columns

Initial Plan For Data Exploration:

Following points summarize the necessary steps to explore the dataset and determine the required processes to ensure dataset are ready to build robust model:

1- Uploading the Dataset
2- Explore the dataset characteristics like (types, distribution, outliers, missing values, relations and correlations …etc)
3- Identifying the proper features
4- List of the required actions for data cleansing and features engineering.
5- Dataset processing
6- Checking the readiness of the dataset.

Actions taken for data cleaning and feature engineering

Checking Null Values:

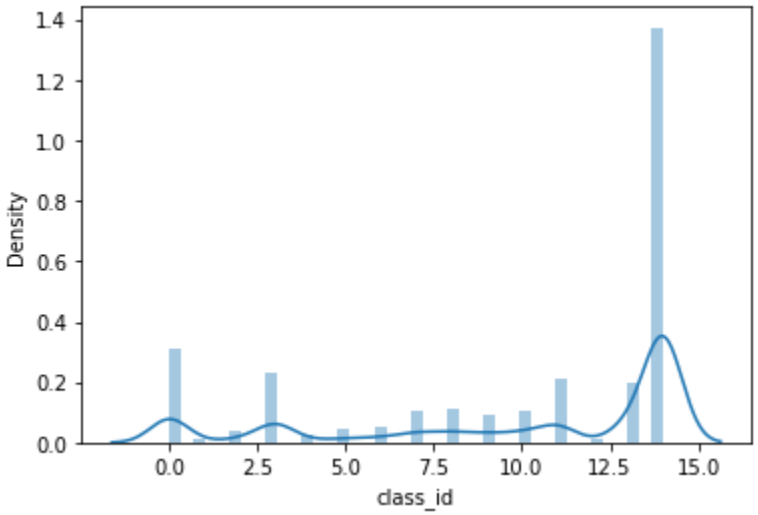
```
In [11]: # checking the null values
         df.isnull().sum()

Out[11]: image_id          0
         class_name        0
         class_id          0
         rad_id            0
         x_min         31818
         y_min         31818
         x_max         31818
         y_max         31818
         dtype: int64
```

Replacing null values with zeros because they belong to the rows(observations) fall under class_id 14 (no diseases) which means there is no boxes(objects) been found in x-ray image

In [15]:
```
# All null values are distributed equivalently between x_min,x_max,y_min,y_max
# All null values fall under class_id 14 (no diseases)
print('The null values: ',df[df['class_id'] == 14].shape[0])

# Replacing null values with zeros because they belong to the rows(observations) fall under class_id 14 (no diseases) which
# means there is no boxes(objects) been found in x-ray image
df.fillna(0, inplace=True)

# Checking the null values:
print('Null values after processing: ', df.isnull().sum())
```

```
The null values:  31818
Null values after processing:  image_id      0
class_name    0
class_id      0
rad_id        0
x_min         0
y_min         0
x_max         0
y_max         0
dtype: int64
```

Dataset looks balanced as number of no diseases observations close the number of diseases observations so there is no need to apply resampling dataset techniques :



In [20]:
```
# Dataset looks balanced as number of no diseases observations close the number of diseases observations so there is no need
# to apply resampling dataset techniques

print('Number of observations belong to no diseases classes: {}'. format(df[df['class_id'] == 14].shape[0]))
print('Number of observations belong to diseases classes: {}'. format(df[df['class_id'] != 14].shape[0]))
```

```
Number of observations belong to no diseases classes: 31818
Number of observations belong to diseases classes: 36096
```

1- **Exclude the observations(rows) of class_id=14 (no diseases) to work on diseases observations** :

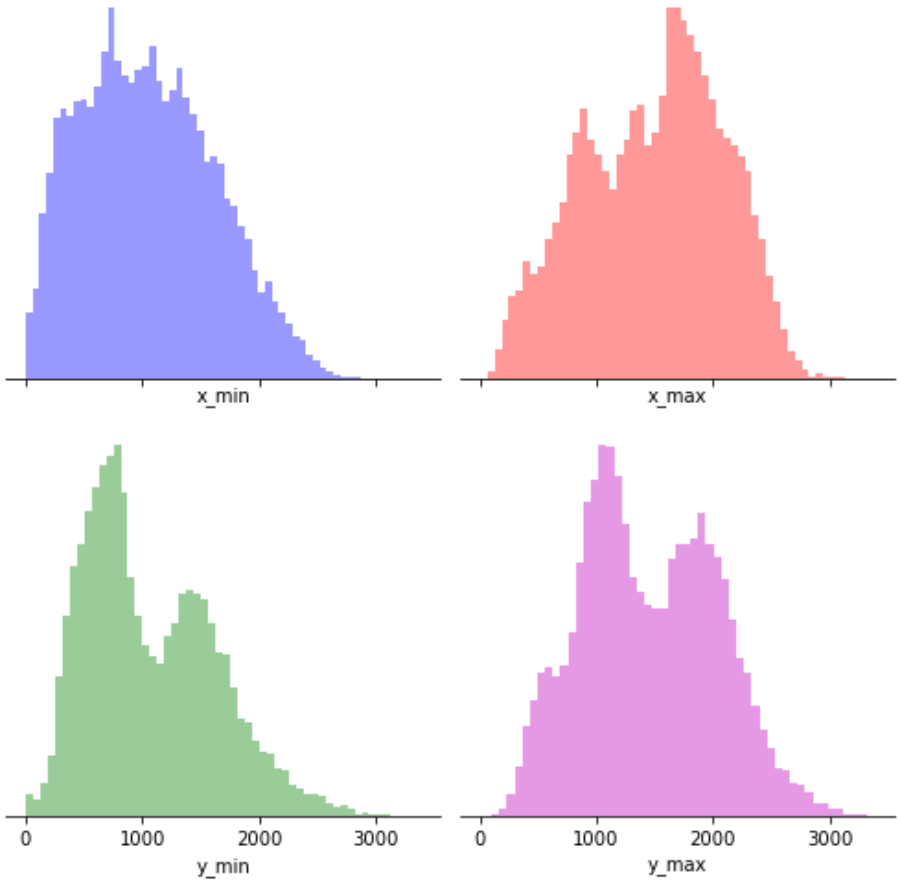
In [16]:
```
# Exclude the observations(rows) for class_id=14 (no diseases)

data= df[df['class_id'] != 14]
data
```

Out[16]:

	image_id	class_name	class_id	rad_id	x_min	y_min	x_max	y_max
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	R10	691.0	1375.0	1653.0	1831.0
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargement	0	R10	1264.0	743.0	1611.0	1019.0
5	1c32170b4af4ce1a3030eb8167753b06	Pleural thickening	11	R9	627.0	357.0	947.0	433.0
6	0c7a38f293d5f5e4846aa4ca6db4daf1	ILD	5	R17	1347.0	245.0	2188.0	2169.0
7	47ed17dcb2cbeec15182ed335a8b5a9e	Nodule/Mass	8	R9	557.0	2352.0	675.0	2484.0
...	...	...	...	...	...	...	...	...
67903	b53d1dd80e99ca6bcef9d592f65d3321	Pleural effusion	10	R9	240.0	1550.0	562.0	2001.0
67906	26d1d5a0ef2e692c6340e74859ffdc53	Pulmonary fibrosis	13	R10	1163.0	787.0	1338.0	941.0
67907	22672ab82c290c20b86863291e25ef6c	ILD	5	R9	299.0	664.0	794.0	1508.0
67908	db169d0be36123bd55b866d6aa73983b	Other lesion	9	R8	6.0	670.0	272.0	1736.0
67912	4b56bc6d22b192f075f13231419dfcc8	Cardiomegaly	3	R8	771.0	979.0	1680.0	1311.0

36096 rows × 8 columns

2- **x-min,x_max,y_min,y_max have fair symmetrical distributions:**

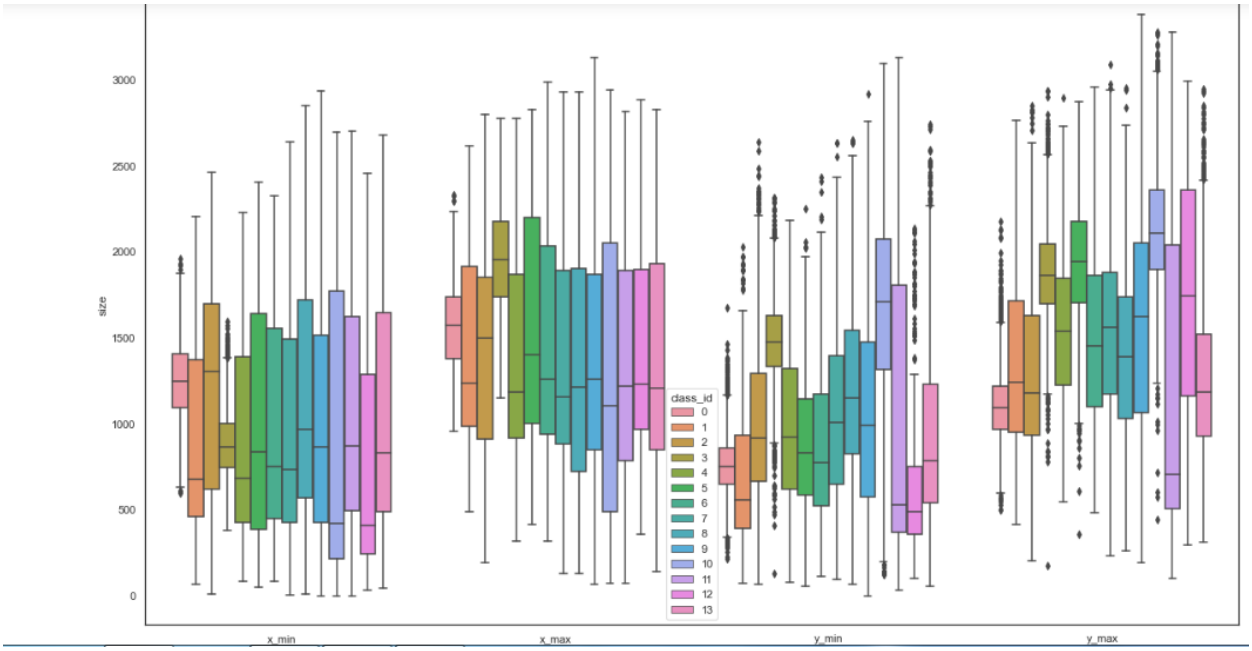


In [42]:
```
# checking the skewness values for x_min,x_max,y_min,y_max
print('The skewness value for x_min is: {}' .format(data['x_min'].skew()), '.......Great,-0.5 < skew value < 0.5 which means x_mi
print('The skewness value for x_max is: {}' .format(data['x_max'].skew()), '.......Great,-0.5 < skew value < 0.5 which means x_ma
print('The skewness value for y_min is: {}' .format(data['y_min'].skew()), '.......Great,0.5 < skew value < 1 which means y_min i
print('The skewness value for y_max is: {}' .format(data['y_max'].skew()), '.......Great,-0.5 < skew value < 0.5 which means x_ma
```

```
The skewness value for x_min is: 0.31593736946368134 .......Great,-0.5 < skew value < 0.5 which means x_min is fairly symmetric
al
The skewness value for x_max is: -0.18676334667868033 .......Great,-0.5 < skew value < 0.5 which means x_max is fairly symmetri
cal
The skewness value for y_min is: 0.5857338744228154 .......Great,0.5 < skew value < 1 which means y_min is moderately  symmetri
cal
The skewness value for y_max is: 0.17911550080779906 .......Great,-0.5 < skew value < 0.5 which means x_max is fairly symmetric
al
```

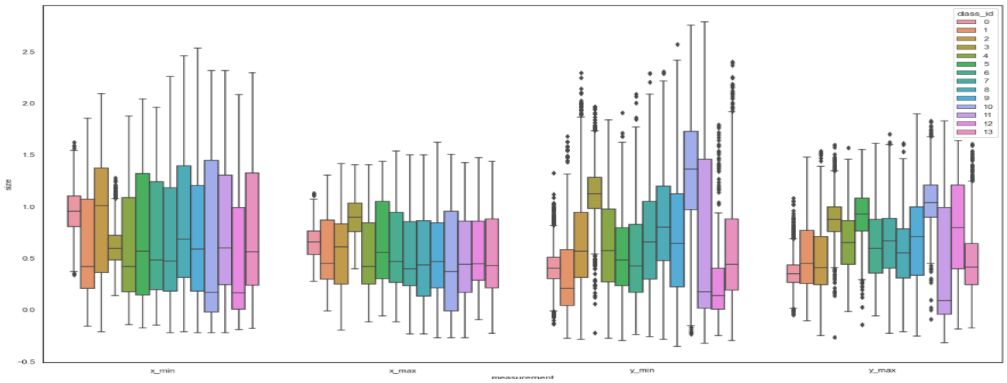
3- Checking the outliers:

In [50]:
```
# Checking outliers in x_min,x_max,y_min and y_max:
data_plot = data[['class_id','x_min','x_max','y_min','y_max']]
data_plot = data_plot.set_index('class_id').stack().to_frame().reset_index().rename(columns={'level_1':'measurement', 0: 'size'})
sns.set_style('white')
sns.set_context('notebook')
sns.set_palette('dark')

f = plt.figure(figsize=(20,12))
sns.boxplot(x='measurement', y='size',
            hue='class_id', data=data_plot)
```

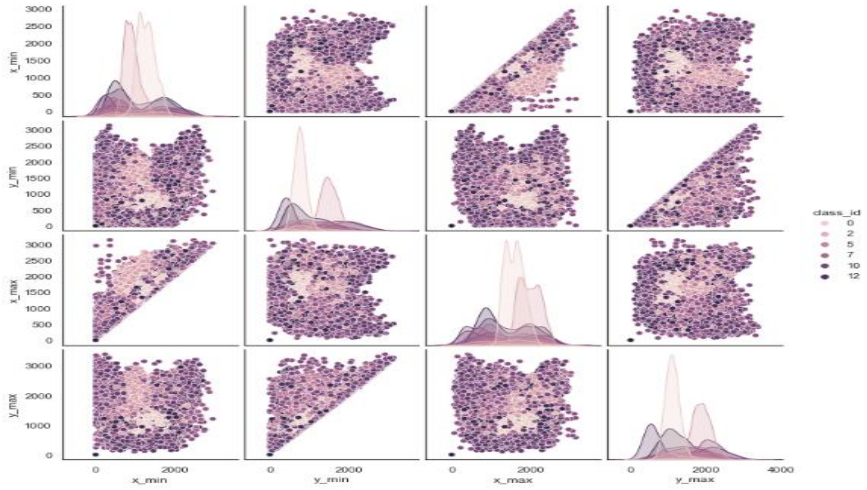


Outliers in y_min and y_max columns is noticeable however outliers are not so extreme .

4- Applying Robust scalar to reduce the outliers, however it looks the outliers numbers and their relative values have not changed! I would prefer to keep the original values as is (since their values are not so extreme) and evaluate their effect on model accuracy (unless a neural model type requires certain type of scaling!):



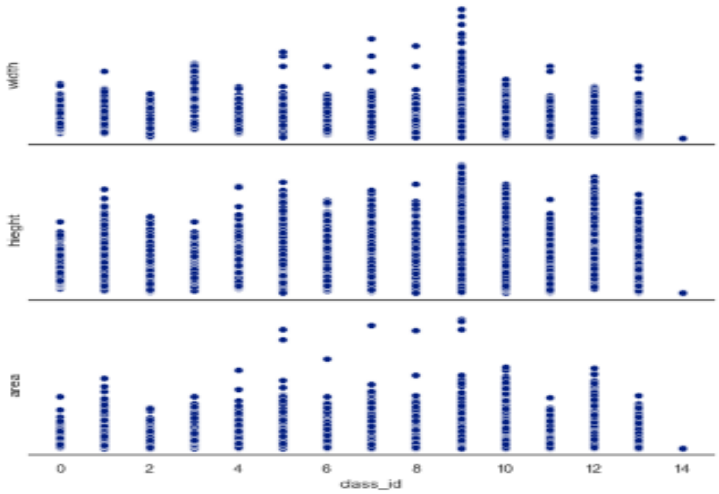
5- Explore the relation and correlation:



6- Adding new features by calculating the width, height and area for each box

In [ ]:
```
df['width'] = df['x_max'] - df['x_min']
df['hieght'] = df['y_max'] - df['y_min']
df['area'] = df['width'] * df['hieght']
df
```

In [117]:
```
f, axes = plt.subplots(3, 1, figsize=(7, 7), sharex=True)
sns.despine(left=True)

sns.scatterplot(data=df,x=df.class_id,y=df.width,ax=axes[0])
sns.scatterplot(data=df,x=df.class_id,y=df.hieght,ax=axes[1])
sns.scatterplot(data=df,x=df.class_id,y=df.area,ax=axes[2])


plt.setp(axes, yticks=[])
plt.tight_layout()
```



Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner

1- Around 50% of values are null which is considered normal and not issue because all these null values are for observations of 'No disease' class, however they been replaced by zero for modeling purpose.
2- The distribution for main features are fair symmetrical (As their skewness values are between -1 and 1) hence no need for any kind of process like log1b or boxcox.
3- There are outliers especially in y_min and y_max columns however their relative values are not so extreme, action has been taken to reduce or eliminate their effect by using Robust scalar however their numbers and relative values have not changed. Decision has been taken to keep the outliers as is and evlaute their effect on model behavior and accuracy, bearing in mind that our dataset are real dataset so these outliers may represent the extreme values and are not considered outliers.
4- New features have been derived from x and y coordinates which are (width, height and area) and we have noticed that width feature is good differentiator.

Hypothesis:

Hypothesis one:

Null hypothesis: Significant number of boxes which width's more than 1500 having Lung Opacity disease.

Alternative: No significant number of boxes which width's more than 1500 having Lung Opacity disease.

Hypothesis two:

Null hypothesis: Significant number of boxes which width's more than 1500 having Cardiomegaly disease.

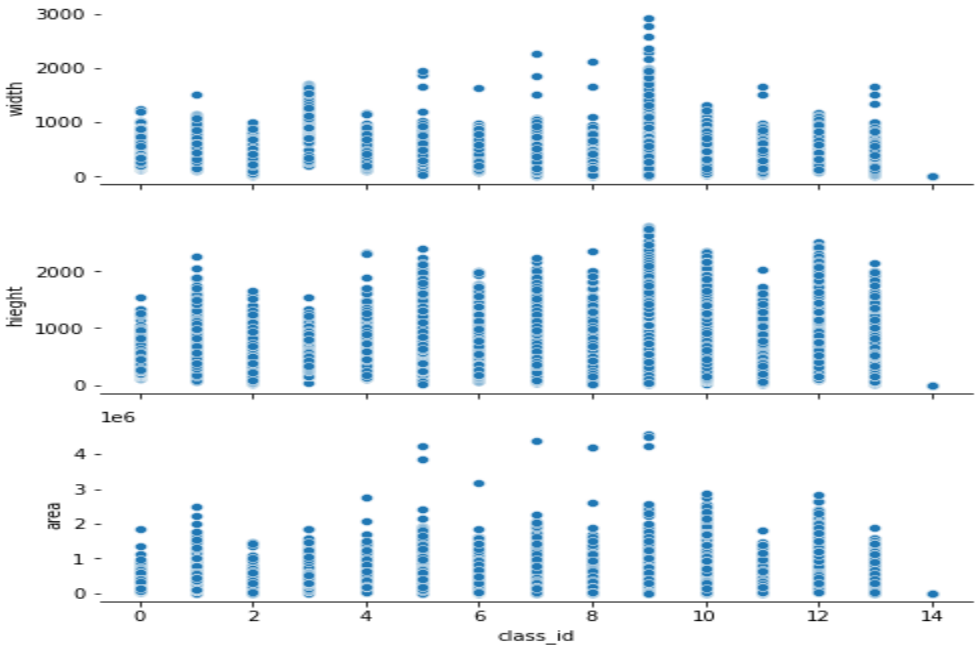
Alternative: No significant number of boxes which width's more than 1500 having Cardiomegaly disease.

Hypothesis three:

Null hypothesis: Significant number of boxes which width's more than 1500 having Atelectasis disease.

Alternative: No significant number of boxes which width's more than 1500 having Atelectasis disease.

Please refer to below plot for more insight:



Hypothesis Testing:

Let us pick up the below hypothesis and conduct significant testing to validate it or invalidate.

Hypothesis:
Null hypothesis: Significant number of boxes which width's more than 1500 having Lung Opacity disease.
Alternative: No significant number of boxes which width's more than 1500 having Lung Opacity disease.

Let us make p-value (cutoff) at 5% and examine the hypothesis :

Using: scipy.stats.ttest_ind
method(https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html#r3566833beaa2-1) to calculate the p-value.

As shown below the p-value is 39% > 5% which means that the probability of finding box of width over 1500 and having lung opacity disease is 39% which more 5% so we can't reject the **null hypothesis**

In [174]:
```
from scipy.stats import ttest_ind


X = df[(df.width > 1500) & (df.class_id == 7)] # filitering the observations (disease is lung opacity and width greater than 1500
Y = df[(df.width > 1500) & (df.class_id != 7)] # filitering the observations (disease is not lung opacity and width greater than


t_stat, p_value = ttest_ind(X['width'],
                            Y['width'])
print("Results:\n\tt-statistic: %.5f\n\tp-value: %.5f" % (t_stat, p_value))
```

```
Results:
        t-statistic: 0.85939
        p-value: 0.39349
```

Suggestions for next steps

Since model not yet been selected so dataset will be more likely subject to more processing based on model requirements and testing findings. Features will be almost subject to certain type of scaling since neural models requires scaled data to work properly so the next challenge is find the appropriate scaling type (If needed)

Data Quality

Dataset quality looks fine and doesn't need intensive processing and transformation however extra process could take place based on model type and testing results.