**About this doc: This doc contains all the commands as and when tried while doing analysis and cleaning. Check the data\_cleaning.sql for the final sql commands used to get the cleaned data.**

SELECT \* into [CEIP].[dbo].[Nest\_Clean]

FROM [CEIP].[dbo].[Nest]

WHERE

dCropUtil != 0 AND dCropUtil IS NOT NULL AND

dPartArea != 0 AND dPartArea IS NOT NULL AND

dTrueArea != 0 AND dTrueArea IS NOT NULL AND

dLengthUsed != 0 AND dLengthUsed IS NOT NULL AND

dWidthUsed != 0 AND dWidthUsed IS NOT NULL AND

cParts != 0 AND cParts IS NOT NULL AND

dArea != 0 AND dArea IS NOT NULL AND

dWidthUsed > 0 AND dLengthUsed > 0 AND

fStrategies IN (0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, -2147483648);

(6758657 rows affected)

SELECT \* INTO Part\_Clean

FROM [CEIP].[dbo].[Part]

WHERE

dLength != 0 AND dLength IS NOT NULL AND

dWidth != 0 AND dWidth IS NOT NULL AND

dArea != 0 AND dArea IS NOT NULL AND

cnested != 0 AND cNested IS NOT NULL AND

dWidth > 0 AND dLength > 0 AND dArea > 0 AND cNested > 0;

(22931580 rows affected)

**In some rows, dLengthUsed and dWidthUsed is more than the dLength and dWidth, so I’m choosing a tolerance of 2 and deleting rows where the deviation is more than this**

DELETE FROM Nest\_Clean

WHERE dLengthUsed > dLength + 2

(20166 rows affected)

DELETE FROM Nest\_Clean

WHERE dWidthUsed > dWidth + 2

(16610 rows affected)

**Setting dLengthUsed = dLength and dWidthUsed = dWidth if difference is less than 2, also setting dTrueArea = dArea (if dLengthUsed = dLength and dWidthUsed = dWidth)**

UPDATE [CEIP].[dbo].[Nest\_Clean]

SET dWidthUsed = dWidth

where dWidthUsed != dWidth and dWidthUsed - dWidth < 2.0 and dWidthUsed - dWidth > 0

(265165 rows affected)

UPDATE [CEIP].[dbo].[Nest\_Clean]

SET dLengthUsed = dLength

where dLengthUsed != dLength and dLengthUsed - dLength < 2.0 and dLengthUsed - dLength > 0

(156744 rows affected)

**Now Altering the Table to set rectified values of dTrueArea and dArea**

ALTER TABLE [CEIP].[dbo].[Nest\_Clean]

ADD dTrueAreaRectified FLOAT;

UPDATE [CEIP].[dbo].[Nest\_Clean]

SET dTrueAreaRectified = dLengthUsed \* dWidthUsed;

(6721881 rows affected)

ALTER TABLE [CEIP].[dbo].[Nest\_Clean]

ADD dAreaRectified FLOAT;

UPDATE [CEIP].[dbo].[Nest\_Clean]

SET dAreaRectified = dLength \* dWidth;

(6721881 rows affected)

**Now Altering the Table to set rectified values of dCroputil**

**[**In ixNest 6296, dCropUtil = dPartArea/dArea

But in ixNest 80719, dCropUtil = dPartArea/dTrueArea**]**

ALTER TABLE [CEIP].[dbo].[Nest\_Clean]

ADD dCropUtilRectified FLOAT;

UPDATE [CEIP].[dbo].[Nest\_Clean]

SET dCropUtilRectified = dPartArea / dTrueAreaRectified;

UPDATE [CEIP].[dbo].[Nest\_Clean]

SET dCropUtilRectified = dCropUtilRectified \* 100;

(6721881 rows affected)

**=========================================================================================**

**Small Analysis: This proves that for around half of the parts dCropUtil is calculated using dArea and for around half it is calculated using dTrueArea**

SeLECt count(\*) from [CEIP].[dbo].[Nest\_Clean]

6721881

SeLECt count(\*) from [CEIP].[dbo].[Nest\_Clean]

where dCropUtil = (dPartArea/dArea)\*100

84675

SeLECt count(\*) from [CEIP].[dbo].[Nest\_Clean]

where ABS (dCropUtil - ((dPartArea/dTrueArea)\*100)) < 2

4277882

SeLECt count(\*) from [CEIP].[dbo].[Nest\_Clean]

where ABS (dCropUtil - ((dPartArea/dArea)\*100)) < 2

5872195

SeLECt count(\*) from [CEIP].[dbo].[Nest\_Clean]

where dArea = dTrueArea

3601170

**=========================================================================================**

**Some columns where dPartArea is greater than dTrueAreaRectified – doesn’t make sense so removing it**

SELECT \* FROM [CEIP].[dbo].[Nest\_Clean]

WHERE dPartArea > dTrueAreaRectified;

I’m getting 10,833 rows

DELETE FROM Nest\_Clean

WHERE dPartArea > dTrueAreaRectified

(10833 rows affected)

**=========================================================================================**

**In Some jobs there are multiple strategies used, so removing them**

Eg.

SELECT \*

FROM [CEIP].[dbo].[Nest\_Clean]

where ixJobSummary = 873526

SELECT ixJobSummary FROM

(SELECT ixJob\_Grouped\_strategy\_wise.ixJobSummary, COUNT(ixJobSummary) AS fStrategies\_Count

FROM

(SELECT ixJobSummary, fStrategies

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP BY ixJobSummary, fStrategies) AS ixJob\_Grouped\_strategy\_wise

GROUP BY ixJob\_Grouped\_strategy\_wise.ixJobSummary) AS ixJob\_Grouped\_strategy\_wise\_With\_Count

where ixJob\_Grouped\_strategy\_wise\_With\_Count.fStrategies\_Count>1

**This returns 203,000 rows, so there are a total of 203,000 jobs which utilizes more than 1 nesting strategy**

**Removing these records**

DELETE from [CEIP].[dbo].[Nest\_Clean]

where ixJobSummary IN (

SELECT ixJobSummary FROM

(SELECT ixJob\_Grouped\_strategy\_wise.ixJobSummary, COUNT(ixJobSummary) AS fStrategies\_Count

FROM

(SELECT ixJobSummary, fStrategies

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP BY ixJobSummary, fStrategies) AS ixJob\_Grouped\_strategy\_wise

GROUP BY ixJob\_Grouped\_strategy\_wise.ixJobSummary) AS ixJob\_Grouped\_strategy\_wise\_With\_Count

where ixJob\_Grouped\_strategy\_wise\_With\_Count.fStrategies\_Count>1)

(1599861 rows affected)

**=========================================================================================**

**Getting number of parts in a job**

From Part table:

Select ixJobSummary, SUM(cNested) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary

ORDER BY ixJobSummary;

From Nest table:

SELECT ixJobSummary, SUM(cParts)

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary

order by ixJobSummary

=========================================================================================

Without matching is cParts in Nest = num of Parts nested

SELECT Parts\_Clean.ixJobSummary, Parts\_Clean.Num\_Parts\_Nested, Nests\_Clean.cParts\_In\_Nest

FROM

(Select ixJobSummary, SUM(cNested) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary) AS Parts\_Clean

INNER JOIN

(SELECT ixJobSummary, SUM(cParts) as cParts\_In\_Nest

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary) AS Nests\_Clean

ON Parts\_Clean.ixJobSummary = Nests\_Clean.ixJobSummary

ORDER BY Parts\_Clean.ixJobSummary

I got 2,050,616 rows

=========================================================================================

With matching is cParts in Nest = num of Parts nested

SELECT Parts\_Clean.ixJobSummary, Parts\_Clean.Num\_Parts\_Nested, Nests\_Clean.cParts\_In\_Nest

FROM

(Select ixJobSummary, SUM(cNested) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary) AS Parts\_Clean

INNER JOIN

(SELECT ixJobSummary, SUM(cParts) as cParts\_In\_Nest

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary) AS Nests\_Clean

ON Parts\_Clean.ixJobSummary = Nests\_Clean.ixJobSummary AND Parts\_Clean.Num\_Parts\_Nested = Nests\_Clean.cParts\_In\_Nest

ORDER BY Parts\_Clean.ixJobSummary

I am getting 1,817,919 rows

**=========================================================================================**

SELECT Parts\_Clean.ixJobSummary, Parts\_Clean.Num\_Parts\_Nested, Nests\_Clean.cParts\_In\_Nest

FROM

(Select ixJobSummary, SUM(cNested\*dArea) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary) AS Parts\_Clean

INNER JOIN

(SELECT ixJobSummary, SUM(dPartArea) as cParts\_In\_Nest

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary) AS Nests\_Clean

ON Parts\_Clean.ixJobSummary = Nests\_Clean.ixJobSummary AND ABS(Parts\_Clean.Num\_Parts\_Nested-Nests\_Clean.cParts\_In\_Nest) < 0.1

ORDER BY Parts\_Clean.ixJobSummary

1,814,342 ABS<10

1,808,482 ABS<1

1,806,107 ABS<0.5

1,804,140 ABS<0.3

1,803,325 ABS<0.2

1,801,522 ABS<0.1

1,798,905 ABS<0.05

1,767,880 ABS<0.01

1,528,169 ABS<0.001

SELECT Parts\_Clean.ixJobSummary, Parts\_Clean.Num\_Parts\_Nested, Nests\_Clean.cParts\_In\_Nest

FROM

(Select ixJobSummary, SUM(cNested) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary) AS Parts\_Clean

INNER JOIN

(SELECT ixJobSummary, SUM(cParts) as cParts\_In\_Nest

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary) AS Nests\_Clean

ON Parts\_Clean.ixJobSummary = Nests\_Clean.ixJobSummary AND Parts\_Clean.Num\_Parts\_Nested = Nests\_Clean.cParts\_In\_Nest

ORDER BY Parts\_Clean.ixJobSummary

1,817,919 rows

**=========================================================================================**

Doing the ABS < 0.1

SELECT Parts\_Clean.ixJobSummary INTO [CEIP].[dbo].[ixJobSummary\_Final\_Area]

FROM

(Select ixJobSummary, SUM(cNested\*dArea) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary) AS Parts\_Clean

INNER JOIN

(SELECT ixJobSummary, SUM(dPartArea) as cParts\_In\_Nest

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary) AS Nests\_Clean

ON Parts\_Clean.ixJobSummary = Nests\_Clean.ixJobSummary AND ABS(Parts\_Clean.Num\_Parts\_Nested-Nests\_Clean.cParts\_In\_Nest) < 0.1

ORDER BY Parts\_Clean.ixJobSummary

(1801522 rows affected)

=========================================================================================

Creating Table with the finalized clean ixJobSummary values: [ixJobSummary\_Final\_numParts]

SELECT Parts\_Clean.ixJobSummary INTO [CEIP].[dbo].[ixJobSummary\_Final\_numParts]

FROM

(Select ixJobSummary, SUM(cNested) as Num\_Parts\_Nested

FROM [CEIP].[dbo].[Part\_Clean]

GROUP BY ixJobSummary) AS Parts\_Clean

INNER JOIN

(SELECT ixJobSummary, SUM(cParts) as cParts\_In\_Nest

FROM [CEIP].[dbo].[Nest\_Clean]

GROUP by ixJobSummary) AS Nests\_Clean

ON Parts\_Clean.ixJobSummary = Nests\_Clean.ixJobSummary AND Parts\_Clean.Num\_Parts\_Nested = Nests\_Clean.cParts\_In\_Nest

ORDER BY Parts\_Clean.ixJobSummary

(1817919 rows affected)

=========================================================================================

Combining [ixJobSummary\_Final\_Area] And [ixJobSummary\_Final\_numParts] into one table

[ixJobSummary\_Final]

SELECT [ixJobSummary\_Final\_numParts].ixJobSummary into [CEIP].[dbo].[ixJobSummary\_Final]

FROM

[CEIP].[dbo].[ixJobSummary\_Final\_numParts]

INNER JOIN

[CEIP].[dbo].[ixJobSummary\_Final\_Area]

ON [ixJobSummary\_Final\_numParts].ixJobSummary = [ixJobSummary\_Final\_Area].ixJobSummary

(1790936 rows affected)

=========================================================================================

**Creating nest and part tables based on this**

select \* into [CEIP].[dbo].[Nest\_Clean\_Filtered]

from [CEIP].[dbo].[Nest\_Clean]

where ixJobSummary in (Select ixJobSummary from [CEIP].[dbo].[ixJobSummary\_Final])

(3803109 rows affected)

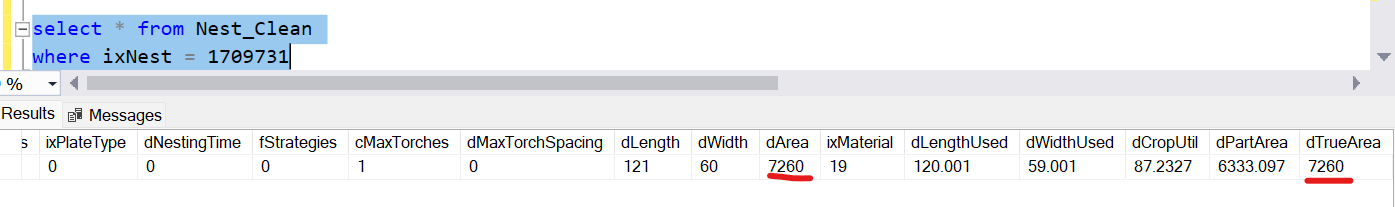
select \* into [CEIP].[dbo].[Part\_Clean\_Filtered]

from [CEIP].[dbo].[Part\_Clean]

where ixJobSummary in (Select ixJobSummary from [CEIP].[dbo].[ixJobSummary\_Final])

(11649316 rows affected)

==================================NOT DOING THIS LOGIC AS IT IS REDUNDANT================ Some cleaning



dTrueArea is same as dArea, but dWidthUsed and dLengthUsed is slightly different, this is causing dTrueArea to deviate from the formula of dLengthUsed \* dWidthUsed

**So setting dLengthUsed = dLength and dWidthUsed = dWidth if dTrueArea = dArea**

UPDATE [CEIP].[dbo].[Nest\_Clean\_Filtered]

SET dWidthUsed = dWidth

WHERE dTrueArea = dArea;

(2618852 rows affected)

UPDATE [CEIP].[dbo].[Nest\_Clean\_Filtered]

SET dLengthUsed = dLength

WHERE dTrueArea = dArea;

(2618852 rows affected)

==================================NOT DOING THIS LOGIC===================================

UPDATE [CEIP].[dbo].[Nest\_Clean\_Filtered]

SET dTrueArea = dArea

where dTrueArea != dArea and dWidthUsed = dWidth and dLengthUsed = dLength

(14759 rows affected)

=========================================================================================SELECT count(\*)

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

where cRequired < cNested

returns 26 – weird but Im ignoring as my aim is to predict cropUtil only, this doesn’t concern me

=========================================================================================

**Queries to evaluate data:**use CEIP

DECLARE @tolerance float;

SET @tolerance = 0.1;

SELECT

(SELECT COUNT(\*) FROM Nest\_Clean\_Filtered WHERE ABS((dCropUtilRectified/100)/ (CAST(dPartArea AS float) / CAST(dTrueAreaRectified AS float)) - 1) > @tolerance)\*100.0 / COUNT(\*) as Percentage\_dCropUtil\_Incorrect,

(SELECT COUNT(\*) FROM Nest\_Clean\_Filtered WHERE ABS(dTrueAreaRectified/(dLengthUsed \* dWidthUsed) - 1) > @tolerance)\*100.0 / COUNT(\*) as Percentage\_dTrueArea\_Incorrect,

(SELECT COUNT(\*) FROM Nest\_Clean\_Filtered WHERE ABS(dAreaRectified/(dLength \* dWidth) - 1) > @tolerance)\*100.0 / COUNT(\*) as Percentage\_dArea\_Incorrect

FROM Nest\_Clean\_Filtered;

**=========================================================================================**

**COLUMNS SELECTED FOR CSV FILE  
[Part\_Clean\_Filtered]**

1. ixJobSummary - use to combine, then eliminate

2. dLength, dWidth, dArea, cNested - avg

3. fExtShape - avg it and see in feature analysis while doing ML (idk how shape can be expressed as a num)

4. dExtArea, dExtBoundaryDist, dExtContainedDist, dLgIntArea, dLgIntBoundaryDist, dLgIntContainedDist, dLgExtConArea, dLgExtConBoundaryDist, dLgExtConContainedDist - avg all of it and do analysis in ml modeling

5. ixPart - just primary key (remove)

6. cRequired - remove (i dont see the use)

7. ixMaterial - remove

**[Nest\_Clean\_Filtered]**

1. ixNest - remove

2. ixJobSummary - use to combine, then eliminate

3. cTimesCut remove (only one value in entire table: 1)

4. fOutput - remove (i dont think it is of any use wrt cropUtil)

5. ixPlateType - remove (not pertinent to CropUtil)

6. cMaxTorches - remove (again not pertinent to cropUtil)

7. dMaxTorchSpacing - remove (not pertinent to CropUtil)

8. dLength, dWidth, dArea, dAreaRectified, dTrueArea - remove (using dLengthUsed, dWidthUsed and dTrueAreaRectified instead)

9. ixMaterial - removing for simplicity of model for now

10. dCropUtil- remove (using dCropUtilRectified instead)

11. cParts - ig remove

12. fStrategies - keep for

13. dLengthUsed - avg it - maybe remove it (see in feature analysis in ML modeling)

14. dWidthUsed - avg it - maybe remove it (see in feature analysis in ML modeling)

15. dPartArea - Sum it for one job

16. dTrueAreaRectified - sum it for one job

17. dCropUtilRectified - remove, once creating table for aggregated jobs, calculate dCropUtil based on aggregate dPartArea and dTrueAreaRectified

**=========================================================================================**

**Generating Final Cleaned Tables**

select ixJobSummary, fStrategies, AVG(dLengthUsed) dLengthUsed\_Avg, AVG(dWidthUsed) AS dWidthUsed\_Avg,

SUM(dPartArea) AS dPartArea\_Job, SUM(dTrueAreaRectified) AS dTrueAreaRectified\_Job

INTO [CEIP].[dbo].[Nest\_Csv]

from [CEIP].[dbo].[Nest\_Clean\_Filtered]

GROUP BY ixJobSummary, fStrategies

(1790936 rows affected)

Select ixJobSummary, AVG(dLength) AS dLength\_Avg, AVG(dWidth) AS dWidth\_Avg, AVG(dArea) AS dArea\_Avg, AVG(cNested) AS cNested\_Avg,

AVG(fExtShape) AS fExtShape\_Avg, AVG(dExtArea) AS dExtArea\_Avg, AVG(dExtBoundaryDist) AS dExtBoundaryDist\_Avg,

AVG(dExtContainedDist) AS dExtContainedDist\_Avg, AVG(dLgIntArea) AS dLgIntArea\_Avg,

AVG(dLgIntBoundaryDist) AS dLgIntBoundaryDist\_Avg, AVG(dLgIntContainedDist) AS dLgIntContainedDist\_Avg,

AVG(dLgExtConArea) AS dLgExtConArea\_Avg, AVG(dLgExtConBoundaryDist) AS dLgExtConBoundaryDist,

AVG(dLgExtConContainedDist) AS dLgExtConContainedDist

INTO [CEIP].[dbo].[Part\_Avg\_Csv]

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

GROUP BY ixJobSummary

(1790936 rows affected)

Select ixJobSummary, dLength, dWidth, dArea, cNested, fExtShape, dExtArea, dExtBoundaryDist,

dExtContainedDist, dLgIntArea, dLgIntBoundaryDist, dLgIntContainedDist,

dLgExtConArea, dLgExtConBoundaryDist, dLgExtConContainedDist

INTO [CEIP].[dbo].[Part\_Csv]

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

(11649316 rows affected)

Select ixJobSummary, MAX(dLength) AS dLength\_Max, MAX(dWidth) AS dWidth\_Max, MAX(dArea) AS dArea\_Max, MAX(cNested) AS cNested\_Max,

MAX(fExtShape) AS fExtShape\_Max, MAX(dExtArea) AS dExtArea\_Max, MAX(dExtBoundaryDist) AS dExtBoundaryDist\_Max,

MAX(dExtContainedDist) AS dExtContainedDist\_Max, MAX(dLgIntArea) AS dLgIntArea\_Max,

MAX(dLgIntBoundaryDist) AS dLgIntBoundaryDist\_Max, MAX(dLgIntContainedDist) AS dLgIntContainedDist\_Max,

MAX(dLgExtConArea) AS dLgExtConArea\_Max, MAX(dLgExtConBoundaryDist) AS dLgExtConBoundaryDist\_Max,

MAX(dLgExtConContainedDist) AS dLgExtConContainedDist\_Max

INTO [CEIP].[dbo].[Part\_Max\_Csv]

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

GROUP BY ixJobSummary

(1790936 rows affected)

Select ixJobSummary, MIN(dLength) AS dLength\_Min, MIN(dWidth) AS dWidth\_Min, MIN(dArea) AS dArea\_Min, MIN(cNested) AS cNested\_Min,

MIN(fExtShape) AS fExtShape\_Min, MIN(dExtArea) AS dExtArea\_Min, MIN(dExtBoundaryDist) AS dExtBoundaryDist\_Min,

MIN(dExtContainedDist) AS dExtContainedDist\_Min, MIN(dLgIntArea) AS dLgIntArea\_Min,

MIN(dLgIntBoundaryDist) AS dLgIntBoundaryDist\_Min, MIN(dLgIntContainedDist) AS dLgIntContainedDist\_Min,

MIN(dLgExtConArea) AS dLgExtConArea\_Min, MIN(dLgExtConBoundaryDist) AS dLgExtConBoundaryDist\_Min,

MIN(dLgExtConContainedDist) AS dLgExtConContainedDist\_Min

INTO [CEIP].[dbo].[Part\_Min\_Csv]

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

GROUP BY ixJobSummary

(1790936 rows affected)

**Generating Nest and Parts Table where num of parts in a job is <=100**

Select \*

INTO [CEIP].[dbo].[Nest\_Csv\_Max100Parts]

FROM [CEIP].[dbo].[Nest\_Csv]

WHERE ixJobSummary IN (

Select ixJobSummary

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

GROUP BY ixJobSummary

HAVING SUM(cNested) < 101);

(1545653 rows affected)

Select \*

INTO [CEIP].[dbo].[Part\_Csv\_Max100Parts]

FROM [CEIP].[dbo].[Part\_Csv]

WHERE ixJobSummary IN (

Select ixJobSummary

FROM [CEIP].[dbo].[Part\_Clean\_Filtered]

GROUP BY ixJobSummary

HAVING SUM(cNested) < 101)

ORDER BY ixJobSummary, dArea DESC;

(6343710 rows affected)

**====================================DONE=================================================**