BHMLAI Capstone project

Author: Bogdan Gavril

March 2023

Executive summary

Business rationale

In the world of hardware design where schedules are extremely tight it is very important to determine the complexity of a project and to estimate the engineering time required for completion ahead of time, usually with limited project details available. In particular, PCB Design projects can range from low complexity to high complexity and the design time varies significantly from project to project. Determining the complexity of a project and the design time have a huge impact on scheduling and on planning the engineering resources. Usually these determinations are made by the engineers themselves or by their managers, but many times there is a significant subjective component which makes these estimations to vary from person to person.

The project

This project proposes the use of Machine Learning methods to make these predictions using existing project data. The outcome consists of Machine Learning models that can be used to predict the design duration and to classify the project as High or Low complexity.
Integrated in project management tools, these Machine Learning models can be used to predict complexity and design time using friendly user interfaces. Project managers, executives and engineers can use the tool for estimating schedules, planning the resources involved in projects or simply classify projects based on their complexity.
The project has two parts.
The first one aims at predicting the complexity of a PCB Design project, labeling it as High or Low complexity.
The second part attempts to predict the PCB Design duration.

Findings

The analysis found that the **best classification model has an accuracy score of 0.76** which is acceptable but must be improved.
The analysis found that the **best prediction model has an accuracy score of 0.42** which is very low. However, we can consider that the accuracy is not critical in this case and make this tool an estimation instrument which can give a suggestion of the design time duration

The analysis also found that the data fails to capture the changes that happen during a design and which can change either the complexity or the design duration or both. For example, a design considered of low complexity and estimated to take 30 days to complete can undergo changes along the way which make it more complex and increase the duration with 20 additional days. The data captures the actual duration of 50 days, but does not include and quantify the changes, thus a 50 days duration is recorded for a set of features which initially indicated a 30 days duration. Considering these, certain steps can be taken to improve the accuracy of the models by changing the way the data is collected and by introducing a method to capture the unplanned changes which affect both the complexity and the design duration. Once the new collected data include these feature, the models can start performing better.

Project: Classification

I used PCB Design projects data collected from various sources and periods of time. The dataset contains 345 entries and 18 features. Below the first five rows are shown.

	line	fit	viatech	viano	pins	layno	sq	dens	dbl	netno	comp	category	ver	complexity	scope	type	duration	level
0	bis	sns	tht	4	20	2	0.21	0.148	dbl	4	6	RG	R2	Medium	MOD-MINOR	FF	73	5
1	ced	mlb	astk	85589	5574	12	71.57	0.180	dbl	728	1454	RG	B0	High	NEW BRD	FF	119	5
2	tgr	mlb	astk	90198	5943	12	80.28	0.189	dbl	814	1541	RG	E0	Medium	MOD-MAJOR	FF	68	5
3	tgr	mlb	astk	62689	5792	12	80.87	0.195	dbl	813	1403	RG	R2	High	MOD-MAJOR	FF	44	5
4	tgr	mlb	astk	63178	5792	12	80.87	0.195	dbl	813	1403	RG	B1	High	MOD-MAJOR	FF	71	5

The dataset features are defined as follows:

Numerical:

- ID - automatically assigned ID number
- viano - number of vias
- pins - number of pins
- layno - number of layers
- sq - layout area
- dens - layout density
- netno - number of nets
- comp - number of components
- duration - layout design duration
- level - engineer expertise level

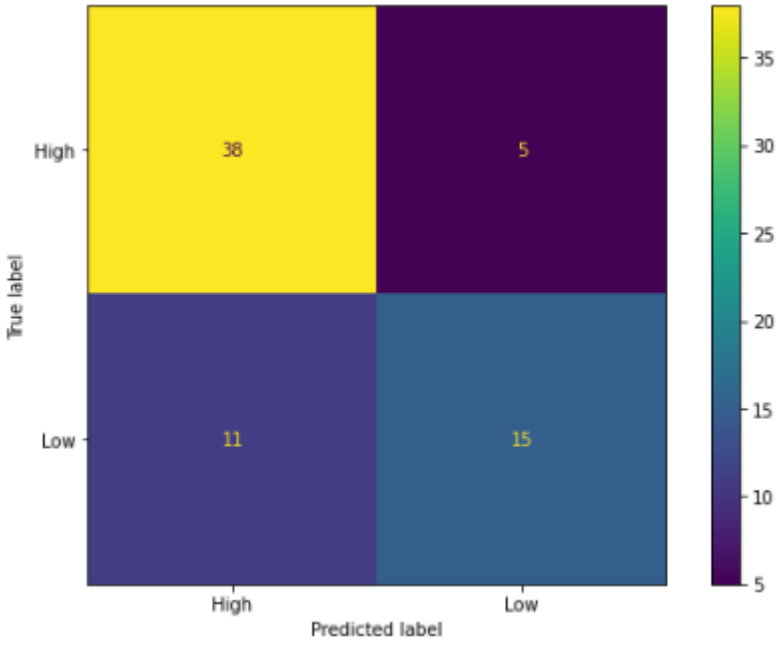
Categorical:

- line - product line (various)
- fit - board utilization (such as daughter board, mlb, etc.)
- viatech - via technology (tht, combinations, etc.)
- dbl - single or double sided
- category - board tech, rigid or flexible
- ver - board version
- complexity - board complexity (low, medium or high)
- scope - scope of the design (new board, minor or major modification)
- type - board family type (such as test, production, etc.)

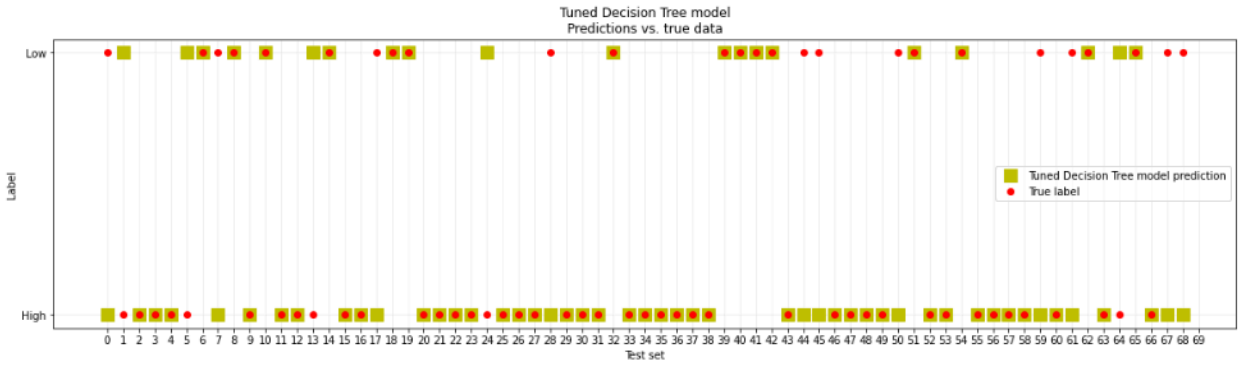
The overall objective is to find the best classification model that can tell whether a project is of high or low complexity.

Findings and results

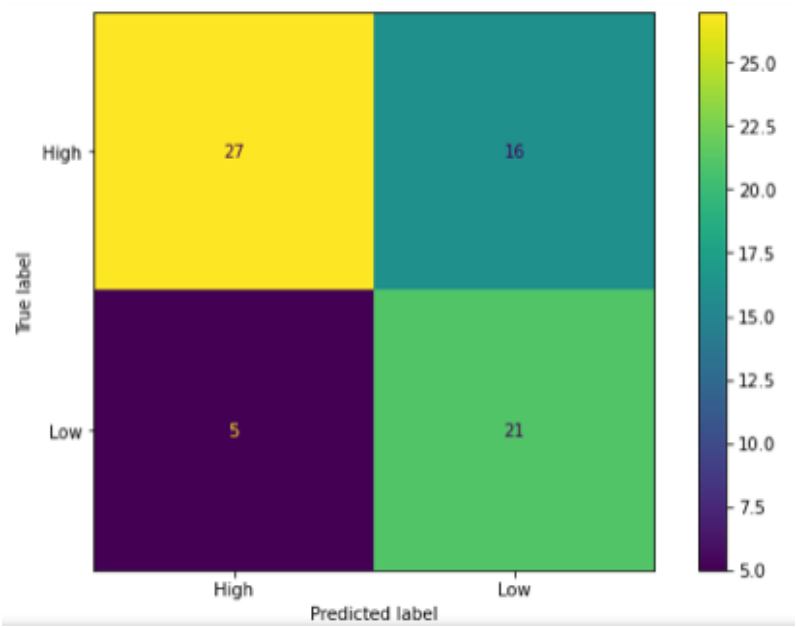
1. The best model is Decision Tree Classifier with an accuracy score of 0.76. A matrix representation of the classifier using the test data set shows that out of the 69 test samples the model classifies correctly 15 'Low' labels and 38 'High' labels and wrongly classifies 11 'Low and 5 'High' labels.

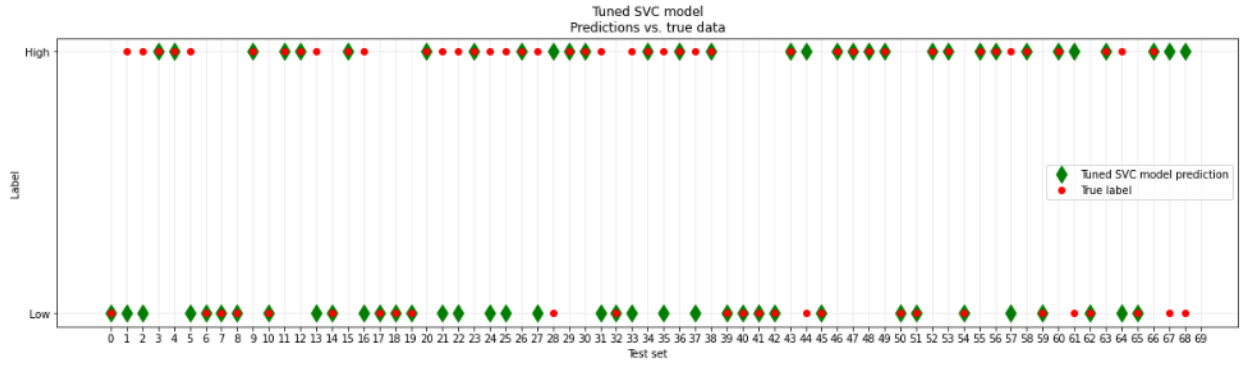


A more visual representation shows the labels which were correctly and incorrectly predicted:



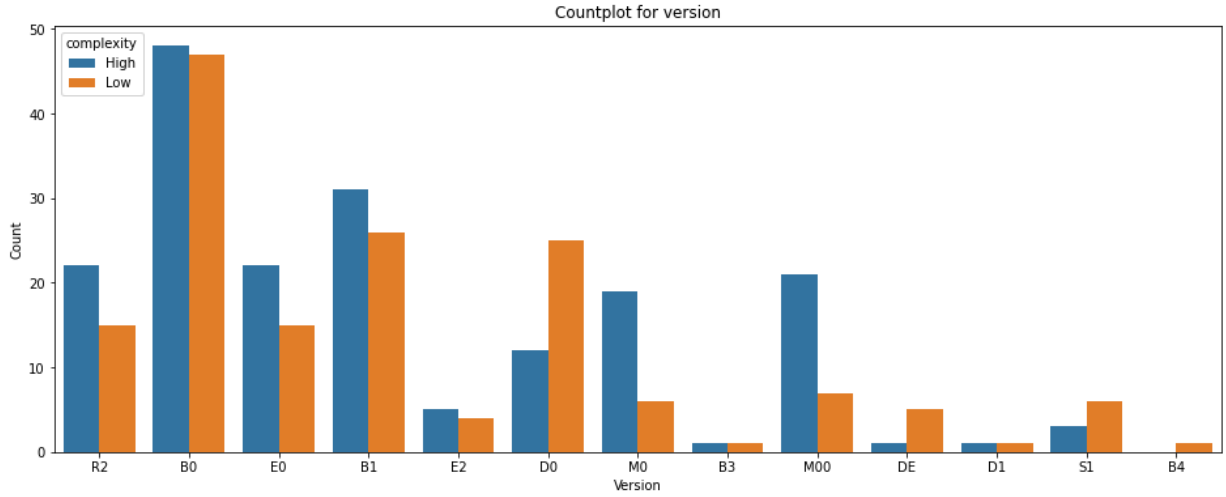
2. The next best model is SVC with accuracy of 0.69. The matrix representation shows that out of the 69 test samples the model classifies correctly 21 'Low' labels and 27 'High' labels and wrongly classifies 5 'Low and 16 'High' labels. The more visual representation shows the correct and the wrong predictions:

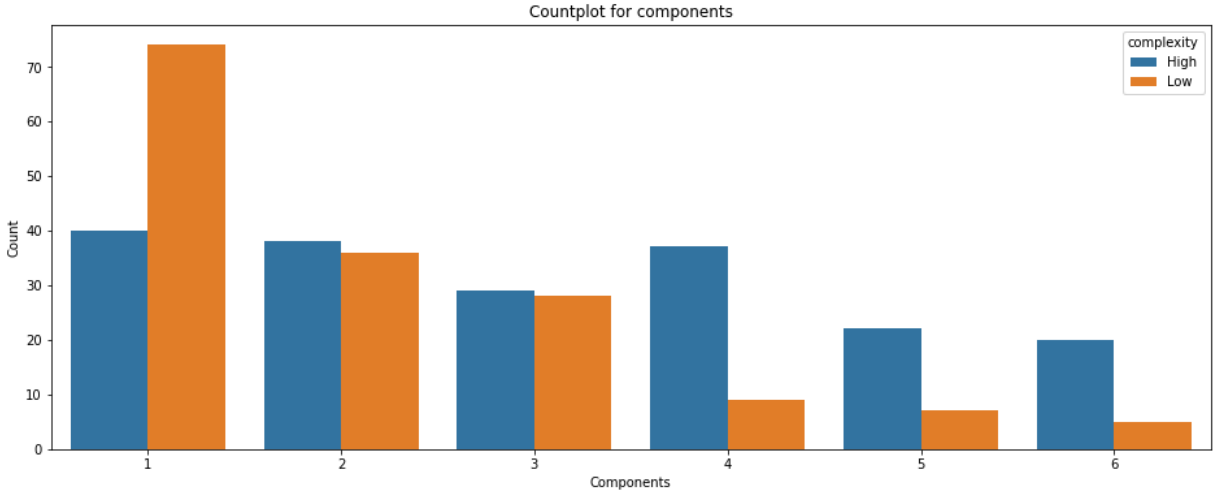


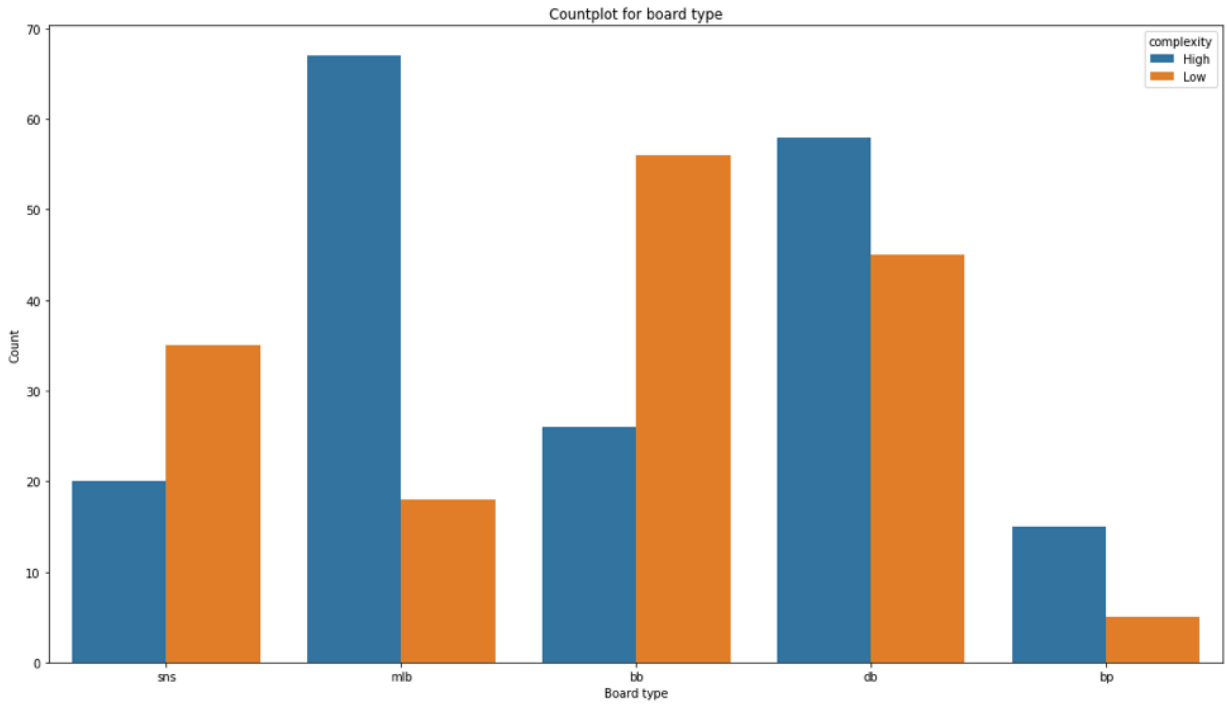


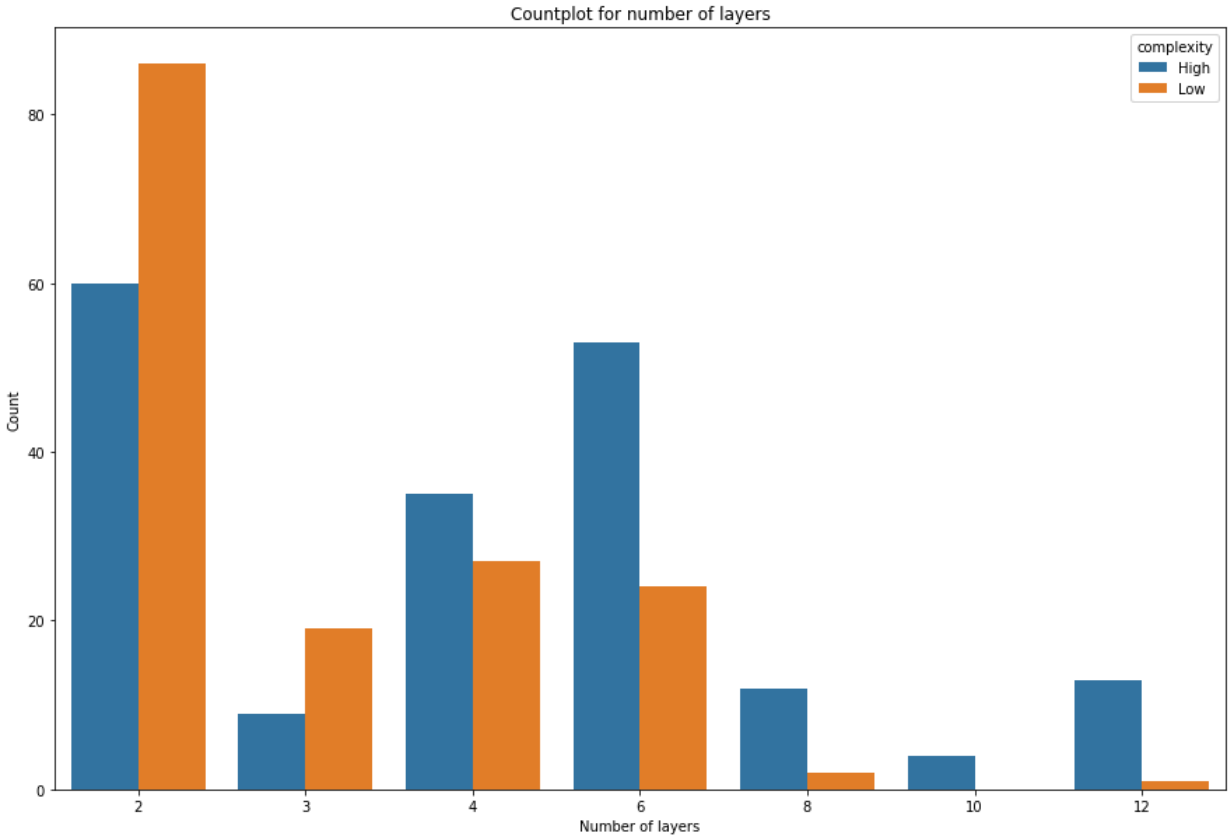
3. Even if the two labels in this dataset are reasonable balanced, the model can classify with a somewhat limited precision
4. The dataset does not have too many entries which limits the training of the models. Also, the determination of the complexity as captured in the dataset was definitely biased, being based on estimation made by engineers or managers, based on subjective factors
5. The dataset and the models cannot capture the unexpected factors which affect the difficulty of a design along the way. A design can start as low complexity based on the initial and change at some point to a highly complex one, in which case the user may update the label, even if the features say otherwise, thus biasing the data. The data I am using fails to capture these variations
6. The classification model can be used as an indicator or an estimation, useful for the business purpose of assigning resources
7. Other useful findings related to the features that can be used as a general guide for project managers, shown in the following plots:

- The complexity depends on the project version
- The more components are on a board, the more complex the design is
- The complexity increases with the number of layers
- The type of board influences the complexity


Countplot for version
complexity
High
Low
Count
Version
R2
B0
E0
B1
E2
D0
M0
B3
M00
DE
D1
S1
B4


Countplot for components
complexity
High
Low
Count
Components


Countplot for board type
complexity
High
Low
Count
sns
mlb
bb
db
bp
Board type



Questions that are addressed

1. *Can I use the available data to build a classifier tool that can help predicting the complexity of a project?*

Yes, the data is reasonable but fails to capture the changes during a project which can affect the complexity of a design

2. *What data can be ignored, or dropped from the analysis?*

Ignore all the features which in practice are not available before the project is finalized

3. *What are the top features that influence the classification?*

`Number of components, scope of design, type of design and product line` have higher importance than others

4. *Is the model accurate enough? What is missing, how can I improve the data set and the analysis as next steps?*

With a precision of 0.76 the model using Decision Tree is reasonable, but there is room for improvement. The first thing missing is more data and some of the next steps are shown below.

Next steps

What can be done to improve the models:

- identify and include additional features which can be collected before the project starts
- collect much more data

What to advise the organization?

- collect more data, historical or as becomes available and reach out to similar organizations for data
- use the classification model with the knowledge that it has a limited precision and can be used in conjunction with human input

How to use this models in practice?

- integrate the models in a practical classification application with an easy to use GUI

Project: Prediction

I used PCB Design projects data collected from various sources and periods of time. The dataset contains 345 entries and 18 features. Below the first five rows are shown.

	line	fit	viatech	viano	pins	layno	sq	dens	dbl	netno	comp	category	ver	complexity	scope	type	duration	level
0	bis	sns	tht	4	20	2	0.21	0.148	dbl	4	6	RG	R2	Medium	MOD-MINOR	FF	73	5
1	ced	mlb	astk	85589	5574	12	71.57	0.180	dbl	728	1454	RG	B0	High	NEW BRD	FF	119	5
2	tgr	mlb	astk	90198	5943	12	80.28	0.189	dbl	814	1541	RG	E0	Medium	MOD-MAJOR	FF	68	5
3	tgr	mlb	astk	62689	5792	12	80.87	0.195	dbl	813	1403	RG	R2	High	MOD-MAJOR	FF	44	5
4	tgr	mlb	astk	63178	5792	12	80.87	0.195	dbl	813	1403	RG	B1	High	MOD-MAJOR	FF	71	5

The dataset features are defined as follows:

Numerical:

- ID - automatically assigned ID number
- viano - number of vias
- pins - number of pins
- layno - number of layers
- sq - layout area
- dens - layout density
- netno - number of nets
- comp - number of components
- duration - layout design duration
- level - engineer expertise level

Categorical:

- line - product line (various)
- fit - board utilization (such as daughter board, mlb, etc.)
- viatech - via technology (tht, combinations, etc.)
- dbl - single or double sided
- category - board tech, rigid or flexible
- ver - board version
- complexity - board complexity (low, medium or high)
- scope - scope of the design (new board, minor or major modification)
- type - board family type (such as test, production, etc.)

The overall objective is to find the best prediction model that can tell what the duration of a new project is.

Findings and results

1. The best model is TransformedTargetRegressor with RandomForestRegressor, with tuned parameters. The models can predict, but with very low accuracy score, even if the Median Absolute Error (MAE) is better than the baseline:

```
Train baseline MAE: 23.66
Test baseline MAE: 23.52
```

Here is a comparison table of all the models used in this analysis:

Comparison Table for models

Model	Train MAE	Test MAE	Score
Linear	17.706953	18.862655	0.252396
Ridge	17.645738	19.155042	0.256456
Lasso	19.278725	19.185290	0.248679
TTR_Ridge	13.322096	14.748382	0.261511
TTR_RFR	5.114385	13.567752	0.422629
TTR_Lasso	19.278725	19.185290	0.248679
Grid_TTR-RFR	5.325523	13.296457	0.295288
Grid_Ridge	17.646523	17.774117	0.205766
Grid_Lasso	19.278725	19.185290	0.189614
Grid_TTR_Ridge	13.987450	15.799061	0.118038
Grid_TTR_Lasso	19.278725	19.185290	0.189614

The table shows that the maximum accuracy score is 0.42. Below Plot 1 is a graphical representation of the prediction produced by the models versus the actual test data and Plot 2 an extract of the predictions produced by the best model.

2. The dataset does not have too many entries which limits the training of the models. Much more data is needed to allow for a good training of the models

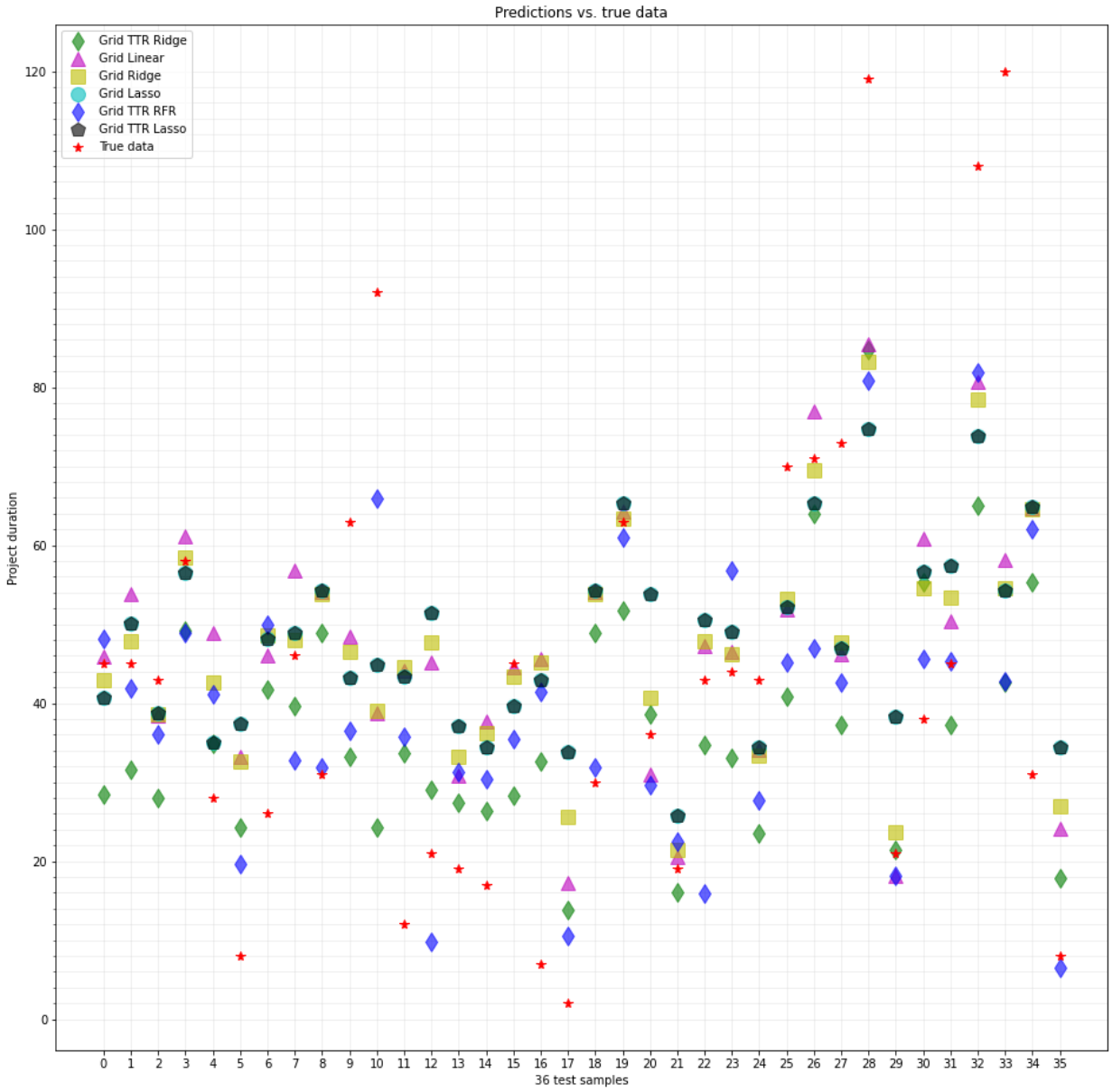
4. The dataset cannot capture the unexpected factors which affect a design along the way. The data has been entered at the end of a project, thus the duration was the real duration as registered at the end of the design. What is important here is that the dataset cannot capture the real events along a project, such as changes and blockers, which delay the design and adds to the overall duration. A design which normally takes 30 days, can be in fact much longer, 60 days or more, due to continuous changes and difficult technical requirements. So, even if the features recorded in the dataset indicate a theoretical duration, in reality the duration can differ vastly.

5. The quality of the data might not be the best, since it was collected with no purpose of this kind in mind.

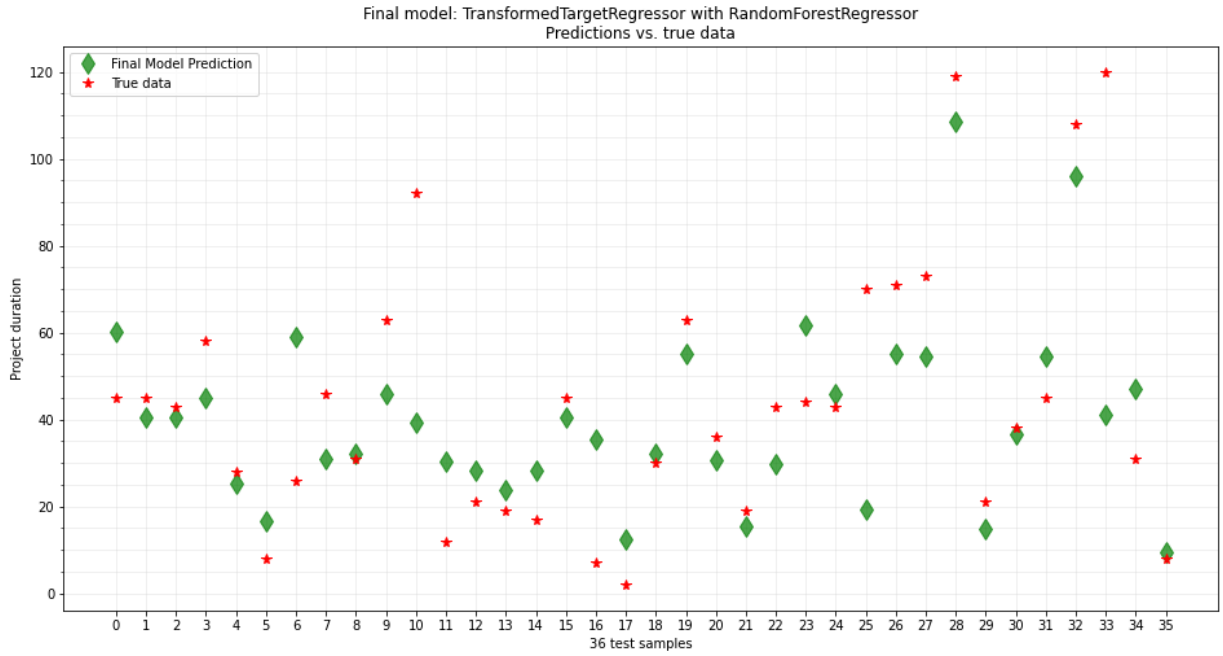
6. The prediction model can be used as a loose indicator, more as an estimation. It can be useful for the business purpose of scheduling, since the results are in a reasonable range. In this particular business case of estimating, low accuracy might be accepted

8. The features which influence the design duration are: `design scope, product line, design complexity, double or single sided, project version`

Plot 1

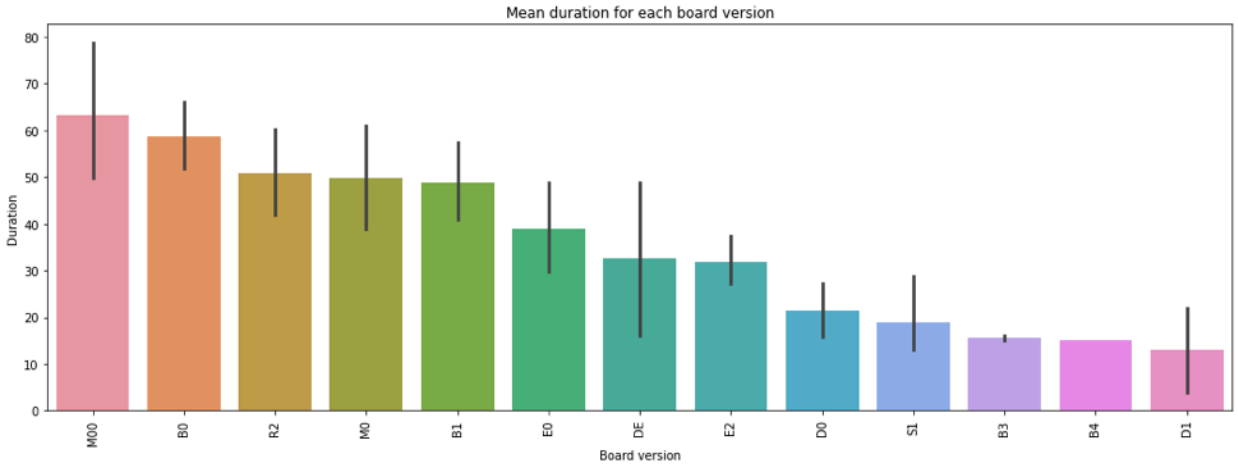

Predictions vs. true data
Grid TTR Ridge
Grid Linear
Grid Ridge
Grid Lasso
Grid TTR RFR
Grid TTR Lasso
True data
Project duration
36 test samples

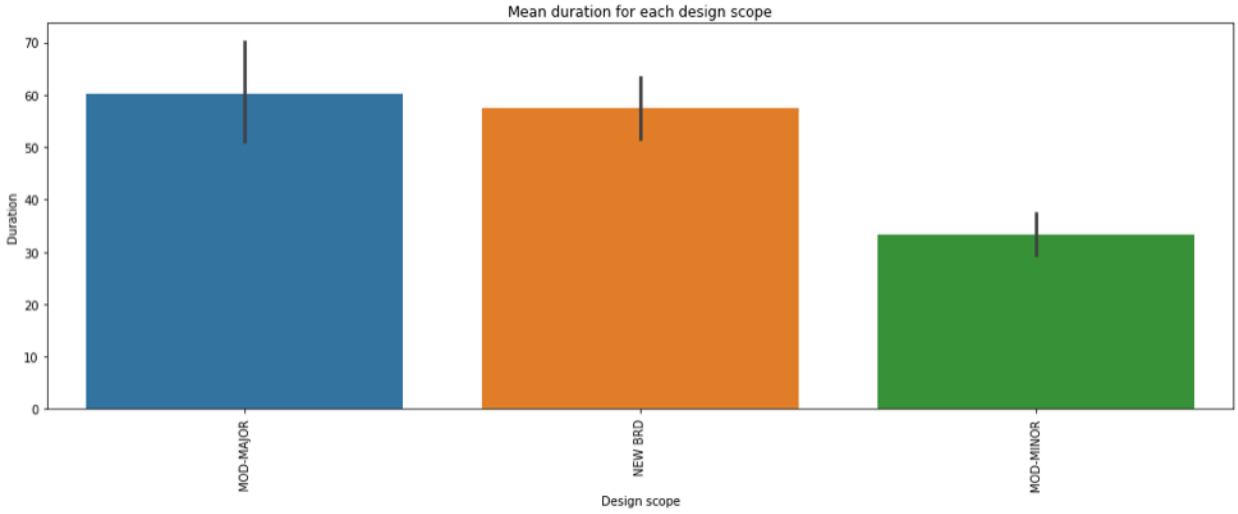
Plot 2

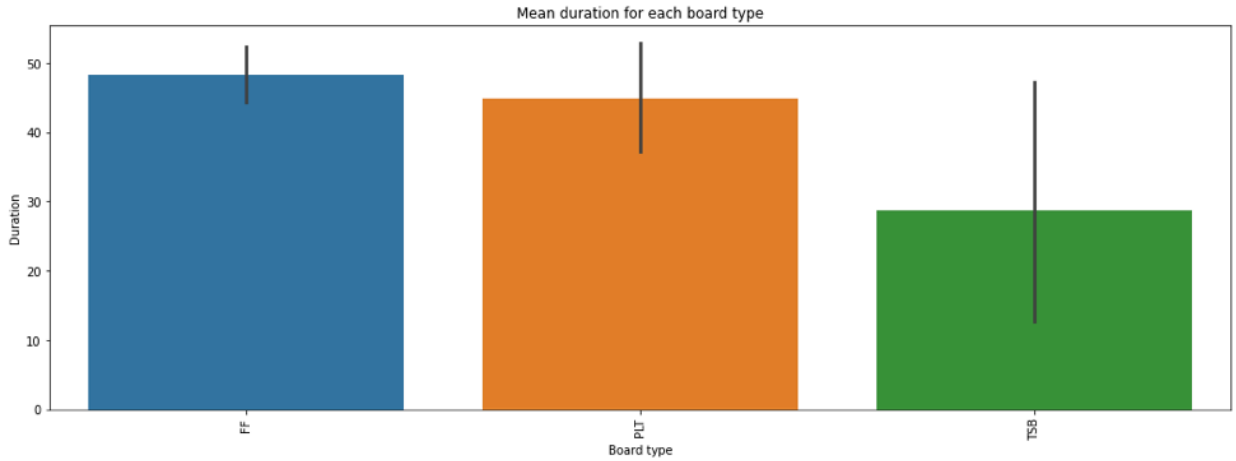


9. Also there are useful findings related to how the features are related to the median project duration:

- Board version has influence over the project duration. Prototype designs take the longest
- Projects take longer when there are new designs or major modifications of previous versions
- Production boards take more time than the flex boards and platform boards







Questions that are addressed

1. *Can I use the available data to build a design duration prediction tool that can predict the duration of a project?*

- Yes, if the accuracy is not a requirement and the results can be used as indicators based on the initial project characteristics

2. *What data can be ignored, or dropped from the analysis?*

- Ignore all the features which in practice are not available before the project is done

3. *What are the top features that influence the duration of a project?*

- `design scope, product line, design complexity, double or single sided, project version` have the highest importance

4. *Is the model accurate enough? What is missing, how can I improve the data set and the analysis as next steps?*

- No, the model is not accurate at all.
- What is missing is much more data to train the models
- I must find a way to capture the missing 'change' and 'subjective' features which directly affect the duration. 'change' should capture a numeric value which represents the amount or percentage of the changes happening along the design. 'subjective' should capture factors such as the teams, the time of the year, vacations or team changes, changes in priorities, etc.

Next steps

What can be done to improve the models:

- identify and include additional features which can be collected before the project starts
- collect much more data
- find a way to quantify and record the changes that happen during a project

What to advise the organization?

- collect more data, historical or as becomes available and reach out to similar organizations for data
- use the prediction model with the knowledge that it has a very low precision and can be used as an indicator and in conjunction with human input

How to use this models in practice?

- integrate the model in a practical prediction application with an easy to use GUI