A Predictive Analysis Model for Incoming Sightings Data

1. Executive Summary

This model predicts future trends and behavior of incoming sightings data, enabling validation of roadmap milestones, roadmap checks and other data-driven management decision making. The paper details the model's mathematical components, and results against historical programs.

2. User Value

Roadmap milestones are based on the program's development level. They can be correlated with count and rate of change of incoming sightings. Historically, it has been observed that milestones align with specific regions of their sightings curve. For example, alpha aligns closely with the initial take-off/high velocity region and beta lies in the latter slow-down/low velocity region. This correlation gives rise to several managerial decision-making use cases.

Determine Risk of Achieving Program Milestones
Use prediction of incoming sightings count to foresee risk to attaining program milestones.

Establish an 'Ideal Sightings Count' curve for tracking program health
Determine a program's 'ideal' sightings curve in order to attain program milestones. Use this ideal behavior as a tracker for teams to follow and deviate minimally from.

3. Introduction

The graph of cumulative (cum.) incoming sightings as a function of workweek (WW) follows an 'S' shape curve. Incoming sightings for a program have low velocity at the initial stage, reach a high velocity in the middle and eventually slow down as the program approaches PRQ and EOL, which creates the curve's 'S' shape. This behavior is summarized as three periods of growth – slow, take-off and rest, which correspond to the mathematical characterizations of growth – exponential, linear and asymptotic respectively. The sigmoid function is a growth model that details these stages and the intricacies a curve exhibits as it transitions from one stage to another.

The prediction model uses sigmoid curve fitting, error analysis, derivative analysis and exponential curve fitting to understand the qualitative behavior of a program's incoming sightings curve, and thereby predict the number of sightings for any future WW.

4. Background

4.1. Stages of Sigmoid Growth Curve

The exponential, linear and asymptotic phases based on the velocity of incoming sightings is evident below -

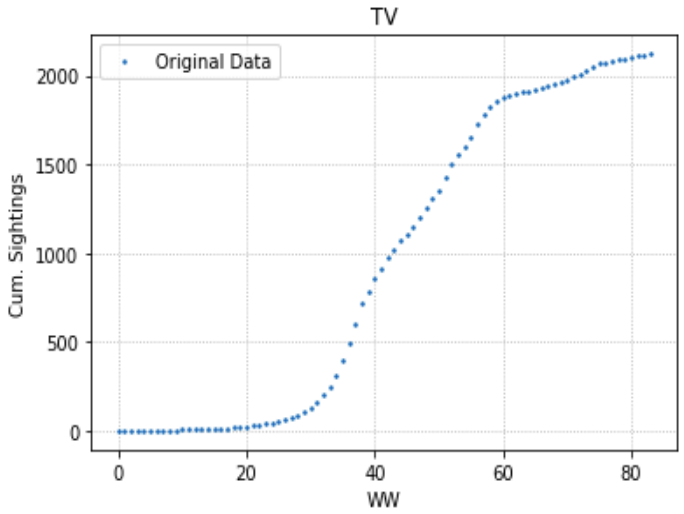


Fig 4.4.1. Original data of TV following sigmoid behavior

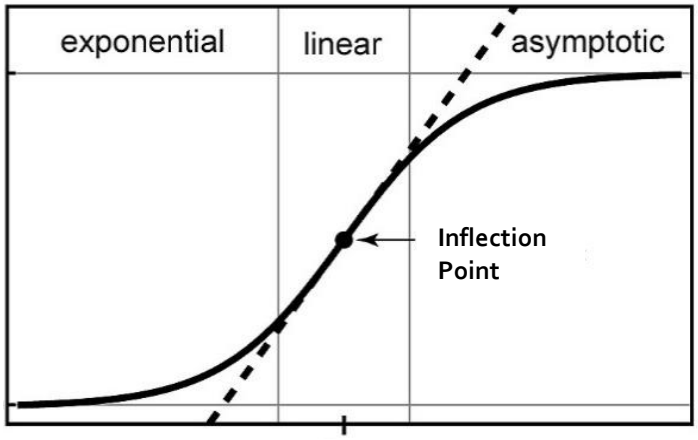
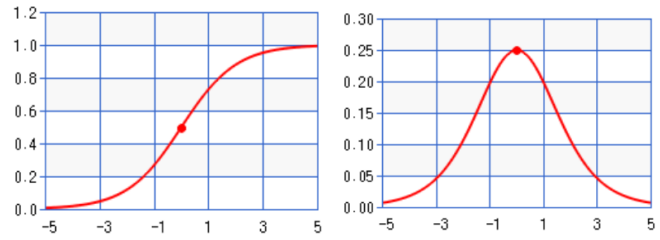


Fig 4.1.1. Phases of sigmoid growth curve

The curve exhibits a slow growth rate in the exponential phase, and transitions to the linear phase where growth is the highest. In the asymptotic phase, growth slows down significantly and reaches an upper plateau or asymptote.

The inflection point (IP) of the curve is the point of symmetry. The top half of the curve from this point on, is a linear transformation of the bottom half of the curve.

4.2. Rate of Change for Incoming Sightings



y = f(x) **y' = f'(x)** - First Derivative

Fig 4.2.1. First and second derivatives of sigmoid function

y'' = f''(x) - Second Derivative

Where, $y = f(x) = \frac{1}{1+e^{-x}}$, the standard unparameterized logistic growth function.

A few key concepts for the prediction model are highlighted below –

C1. Sigmoid curves contain exponential, linear and asymptotic phases corresponding to velocity of incoming sightings.

C2. The x-value (weeks) at **max**(first derivative, f'(x)) is the inflection point of the curve.

C3. The region between the **max**(second derivative, f''(x)) and **min**(second derivative, f''(x)) corresponds to the linear phase of the curve.

C4. Furthermore, the **median**(x-value) in the linear phase of the sigmoid curve is the inflection point.

For a new program, using exponential curve fitting, the beginning of the linear phase is identified (C1.). Using historical data about linear phase lengths (LPL), an estimation for the new program's LPL is made based on its size (C3.). Once the inflection point of the program is reached, a linear transformation of the curve from this point onwards is a prediction of the sightings trend (C4.).

5. Model

5.1. Historical Data Analysis (Completed programs)

5.1.1. Data

Incoming sightings data is extracted directly from the Jira database using the following query –

> "project in ("NSG System Engineering")
> and program in (*Program Name*)
> and issuetype = sighting"

This data is truncated for duration = PRQ 1 + 26 weeks. A cumulative function count is applied to find the total number of tickets for each WW. This data serves as the primary input for the model.

5.1.2. Sigmoid Curve Fitting

Sightings data for a program is fitted to a four-parameter sigmoid function using a non-linear least squares regression optimization method. After curve fitting, a parameter set is returned that details the curve's upper plateau, bottom plateau, inflection point and slope. The parameterized sigmoid function is given by -

$$y = f(x) = \frac{a}{1 + e^{(-c(x-d))}} + b \qquad \text{Eqn. (1)}$$

For Fultondale (FD, Data Center PCie program), sigmoid curve fitting results in parameters –

a = 2937.196368 (Top plateau)
b = 2.597591 (Bottom plateau)
c = 0.068386 (Measure of slope)
d = 96.822604 ~= 97 weeks (x-value - weeks at inflection point)
Avg. Residuals = 17 sightings (Difference between true sightings counts and fitted data)

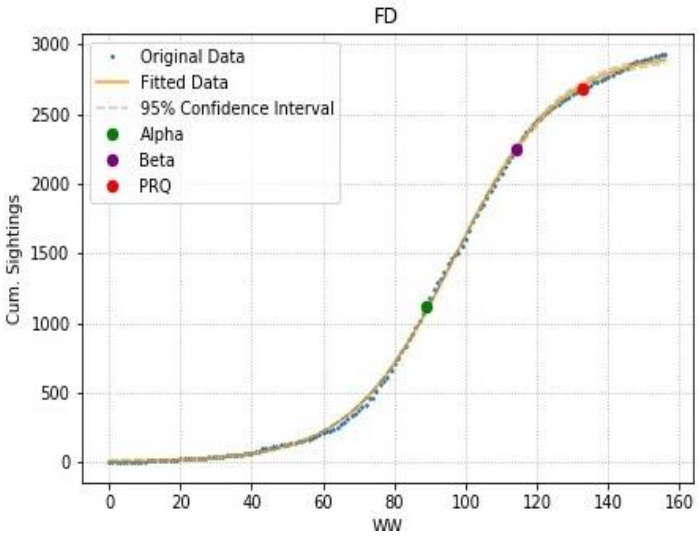


Fig 5.1.1. Sigmoid curve fitting for FD

The goodness of fit, measured by average residual values, for curve fitting on 18 programs, proves that incoming sightings generally follow sigmoid behavior. This assumption is fundamental to the prediction model.

5.1.3. First & Second Derivative Analysis

The first and second derivative is computed to understand rate of incoming sightings. The linear phase of the curve is isolated using the maximum and minimum of second derivative.

For FD, first derivative and second derivative computations result in –

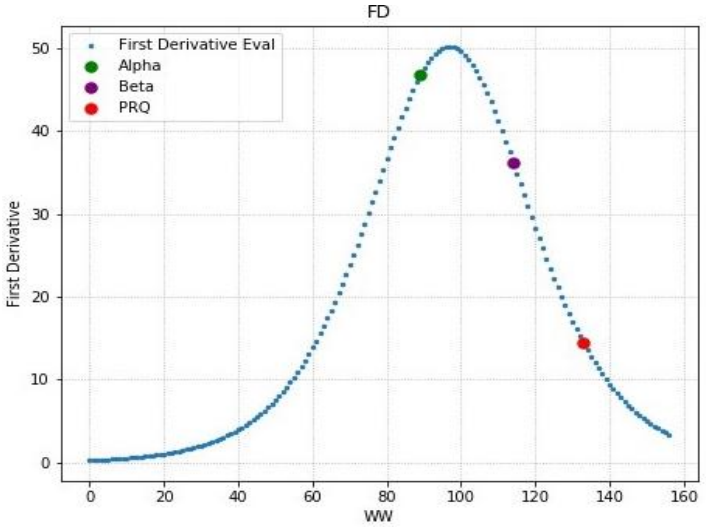


Fig 5.1.2.1. First derivative plot for FD

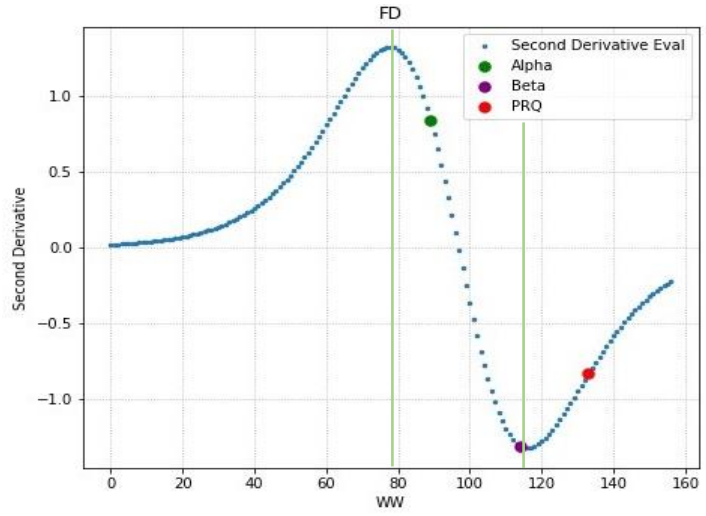


Fig 4.1.2.2. Second derivative plot for FD

Linear Phase Length (LPL) for FD =

(weeks at **max**(f''(x))) – (weeks at **min**(f''(x)))
abs(79 weeks - 117 weeks) = 38 weeks

5.1.3. Correlation to Program Milestones

Alpha

The alpha milestone correlates with the region in the fitted sigmoid function where exponential phase ends and linear phase begins. Since teams begin rapidly working on the program at this stage, a high rate of change in incoming sightings counts is observed. Thus, alpha ideally aligns with the period where the second derivative is maximum.

Beta

The beta milestone correlates with the region in the fitted sigmoid function where linear phase ends and asymptotic phase begins. Since teams are getting closer to PRQ, the rate of change of incoming sightings decreases significantly. Thus, beta ideally aligns with the period where the second derivative is minimum.

PRQ

The PRQ 1 milestone typically occurs a few weeks after the program hits beta. Since teams are getting closer to releasing the product, incoming sightings rate falls significantly. Thus, PRQ ideally aligns with the asymptotical phase of the fitted sigmoid curve.

For FD, alpha occurs at week 89 – soon after the **max**(f''(x)) which is at 79 weeks. This is the take-off period where the program is ramping up. Beta is at week 114 - very close to **min**(f''(x)), which is at 117 weeks. PRQ is at week 133 – 19 weeks after beta.

5.1.4. Approximating LPL Using Program Size

Program size is measured using its maximum cum. sightings value. A relationship between LPL and program size is drawn as a result.

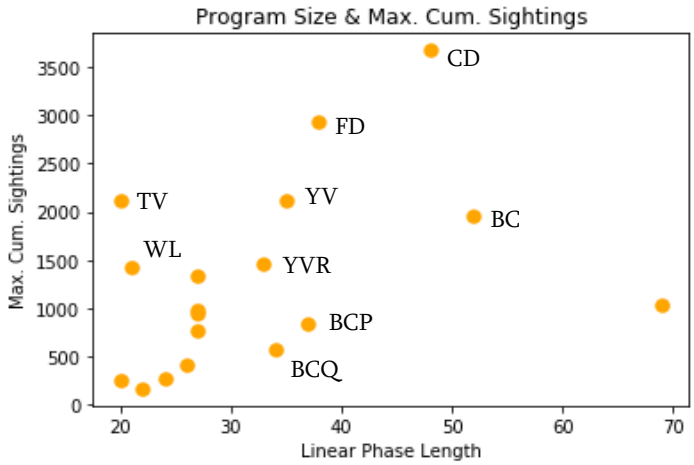


Fig 5.1.4.1. Program Size & Max. Cum. Sightings

Program Size is meaured by its max. cum. sightings count. Derivative programs are clustered at LPL range of 20-40 weeks.

LPL and Program Size are correlated as follows –

Program Size	Linear Phase Length
Small	30 weeks
Medium	40 weeks
Large	50 weeks
Extra Large	60 weeks
***Extra Large**	70 weeks

5.2. New Program Data Analysis & Prediction

5.2.1. Exponential Curve Fitting

Exponential curve fitting is used to identify the beginning of linear phase and hence the inflection point of a new program. Starting with a subset of available data points, a parameterized exponential curve is fitted to the values.

$$y = f(x) = a * e^{(-b(x-c))} + d \qquad \text{Eqn. (3)}$$

Without changing the parameters of the fit, the sightings data for the next ten weeks is observed. If there is significant deviation from the fit, it is argued that exponential phase has stopped, and data has entered linear phase. If the deviation is negligible, data continues to be in exponential phase and a check for the LP transition can happen again in 10+ weeks. This process is iterated until the beginning of LP is attained.

For FD, after 5 such iterations the start of LP is found -

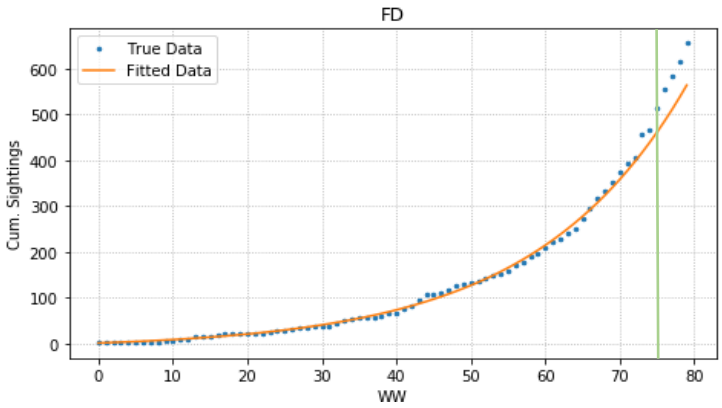


Fig 5.2.1. Exponential curve fitting to identify LPL

Significant deviation from the exponential curve is observed in week 75 (Residual > 50 sightings). Thus, for this week onwards, the curve is said to be in the linear phase. In essence, week 75 marks the beginning of linear phase and end of exponential phase.

5.2.2. Applying LPL & Identifying Inflection Point

Using a known relationship between program size and linear phase length from Section 5.1.4, the LPL value of a new program is approximated. To account for variations, 4 bounds - +33%, -33%, +16.5% and -16.5% are applied on the base LPL value.

FD is considered a medium sized program, so a base LPL approximation of 40 weeks is applied (Section 5.1.4).

The inflection point (point of symmetry) is the median of the linear phase. The weeks value and sightings count at this point are computed. Once the program's data has reached the IP, a linear transformation of pre-IP data becomes a prediction for future values (post-IP).

This process is iterated for each LPL bound (+33%, -33%, etc.), generating 5 IPs and hence 5 prediction curves.

5.2.3. Applying Linear Transformations

The pre-IP data points are laterally and vertically inverted. The original and transformed data are concatenated to make the final prediction curve. This process is repeated for each LPL bound.

For example, in FD a linear transformation on pre-IP data (LPL = 40 weeks) –

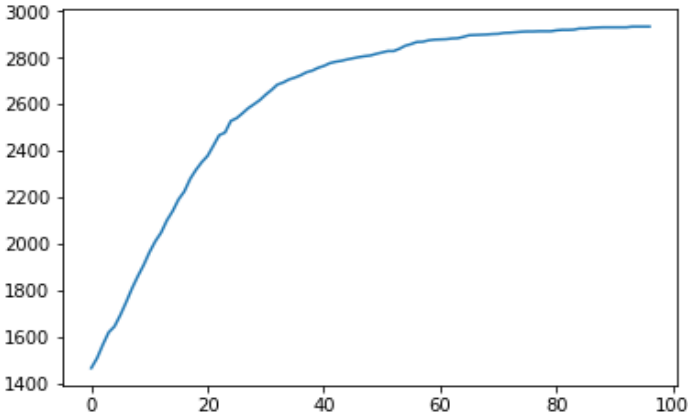
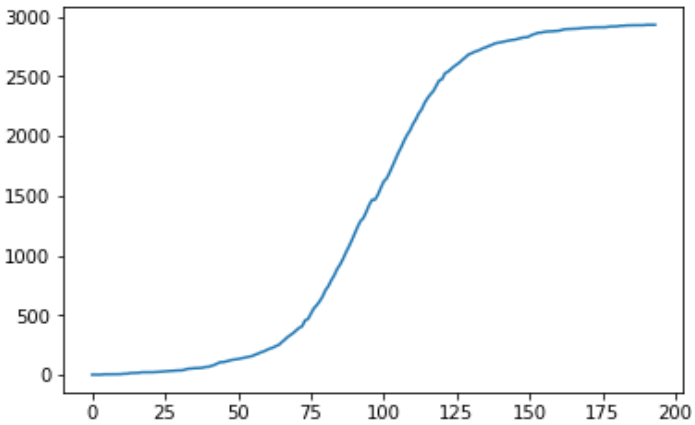


Fig 5.2.3.1. Vertical and lateral inversion of pre-IP data

Concatenating pre-IP and post-IP values –



5.2.3.2. Concatenated pre-IP and post-IP data

5.2.4. Identifying Best Prediction Using Residual Error Analysis

A range of curves are generated using the method described in sections 4.2.2 and 4.2.3 for each LPL (bound) and corresponding inflection point.

The best prediction is the curve with least residual error when compared to the true data.

For FD, this is the curve with LPL + 16.5%.

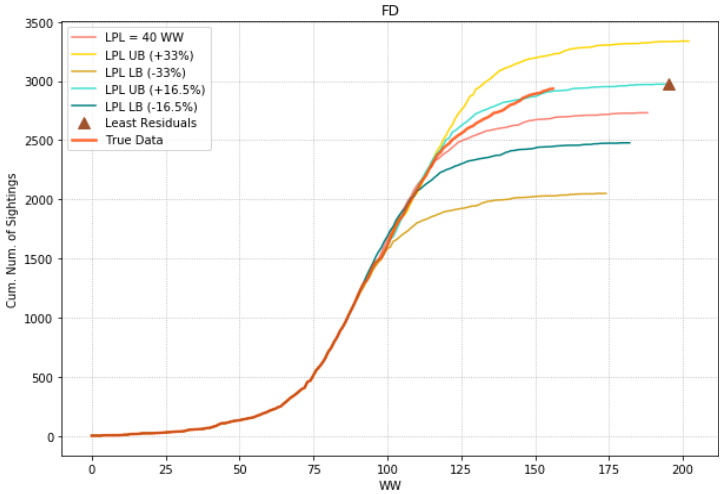


Fig 5.2.4.1. Prediction curves for FD

Determiners	True	Predicted
Week beginning linear phase	79 weeks	75 weeks
Inflection Point (IP)	98 weeks	97 weeks
Linear Phase Length (LPL)	38 weeks	46 weeks
Week ending linear phase	117 weeks	121 weeks

5.2.5. Step-by-Step Guide (Summary for FD)

The prediction model is applied after 20-30 weeks of FD's operation – period where the program is showing movement and approaching take-off.

An exponential fit is done on 30 weeks of sightings data, which is incremented by 10 weeks until a significant deviation from the fit is observed. For the first four fits, deviation is lesser than 50 sightings, which is negligible. For the fifth iteration, a deviation >50 sightings is observed at week 75. Hence, from week 75 onwards the data is in linear phase.

The approximations for LPL (Section 5.1.4), indicate that FD's base LPL value is 40 weeks. The middle of LP is given by 75 + median(40) = 95 weeks, which is the inflection point.

This process is iterated for the following LPL bounds - LPL Base, LPL + 0.33*LPL, LPL - 0.33*LPL, LPL + 0.16*LPL, LPL - 0.16*LPL. Each bound produces a unique IP.

When 95 data points are available, the pre-IP data set is complete. The data pre-IP is inverted L-R and U-D, creating the predicted data for post-IP. Concatenating pre-IP (bottom half of curve) and post-IP (upper half of curve) datasets, creates the prediction curve.

This process is repeated for each IP value, generating five prediction curves.

5.2.6. Results

The model is applied to 18 completed programs to test the closeness of predicted data to true values. Overall, the prediction is stronger for larger programs as more data points are available to analyze.

6. Challenges

High quality data is essential for precision and optimization in curve fitting methods. Some programs have data that show abnormal deviations which creates poor fitting parameters, which consequentially affect first and second derivative values, LPL approximations, etc.

Furthermore, assuming that the sightings curve is sigmoidal implies that there is symmetry about the inflection point, which is critical to the prediction model. However, in some cases, curves do not have precise sigmoidal behavior. This leads to over-approximations or under-approximations in predictions. It can be accounted for by improving error bounds using further historical data analysis, to provide more accurate 'landing zones' for predictions.

In addition, the methodology to report sightings might change over time, affecting the ability to compare datasets across programs. This impacts LPL approximations and consequentially the prediction for a new program, as the predicted LPL is based on historical programs with similar size. This can be mitigated by adding or deleting categories of sightings to ensure that datasets are universal.

7. Current Prediction for ADP (9.20.2019)

The prediction for ADP (Arbordale Plus) states that the true data is trending towards the region in between curves – LPL+16.5% and LPL+33% (inclusive).

These two bounds provide ranges for the determiner values.

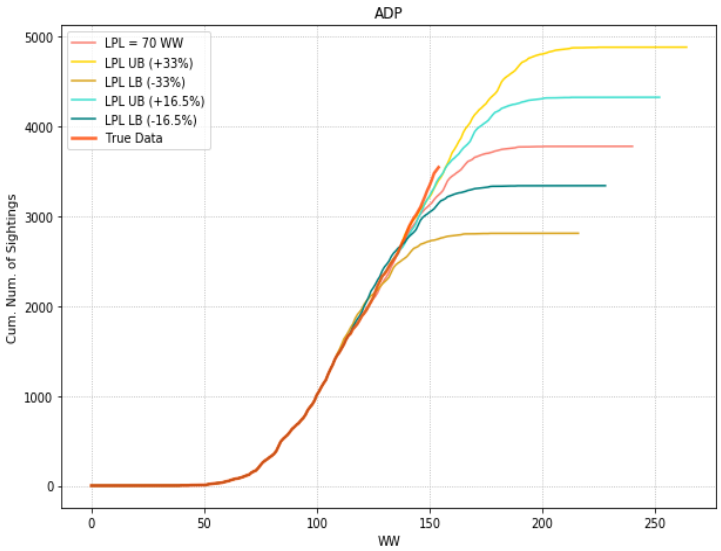


Fig 7.1. ADP Predictions

Determiners	True	Predicted	
		+16.5%	+33%
Week beginning linear phase	N/A	86 weeks	86 weeks
Inflection Point (IP)	N/A	126 weeks	132 weeks
Linear Phase Length (LPL)	N/A	81 weeks	93 weeks
Week ending linear phase	N/A	167 weeks	179 weeks

A base LPL value of 70 weeks (Section 5.1.4.) is applied.

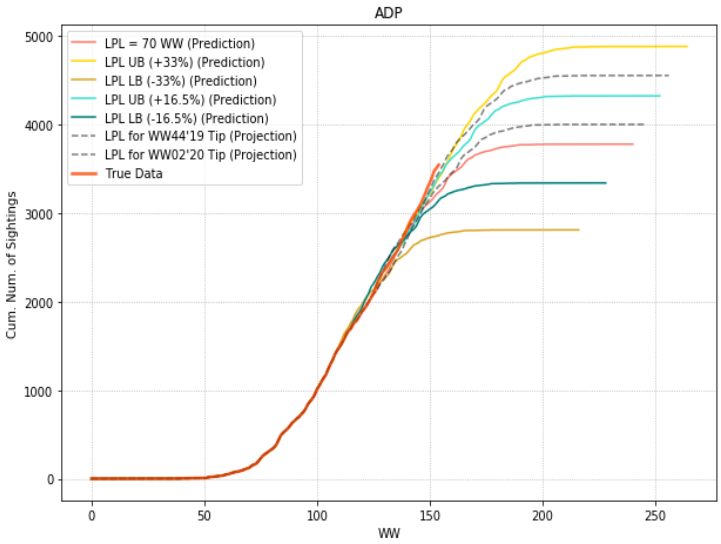


Fig 7.2. ADP Predictions & Projections

To determine sightings counts as goals to reach, in order to achieve beta in a particular week (slow down, ending LP), projection curves can be drawn by implementing the model backwards. Here, no historical programs are used to estimate LPL. The LPL approximation comes from working in reverse using the desired ending of LP.

For example, for ADP if WW 02' 2020 is beta (161 weeks from the beginning of the program),
LPL = 161 weeks – (Weeks at beginning of LP)
LPL = 161 – 86 = 75 weeks.

Using this LPL value, a projection curve is drawn for the beta target - WW 02' 2020. This provides sightings counts for groups to track and measure their closeness to reaching the respective milestone.

8. Python Implementation

Model is implemented in Python using NumPy, Pandas, Scikit, SymPy modules for curve fitting, optimization, derivative calculation, etc. Running the scripts in an Anacondas IDE is ideal as all the required modules are pre-installed.
Scripts and further documentation accessible here -

9. Further Applications

The methodology for prediction is applicable to any data set that follows sigmoid growth behavior. There is no implicit dependency on sightings specific data for the model's functionality. The corresponding Python code can be restructured to read non-sightings data sets.

References

1. Revisiting the Estimation of Dinosaur Growth Rates - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Three-phases-of-growth-in-a-typical-sigmoidal-curve-A-sigmoidal-curve-solid-black-line_fig9_259395938 [accessed 5 Sept. 2019]

2. A Novel Hybrid Classification Model of Genetic Algorithms, Modified k-Nearest Neighbor and Developed Backpropagation Neural Network - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Graph-of-the-Logistic-function-and-its-derivative-function_fig1_268874045 [accessed 10 Sept. 2019]

3. Curve fitting using non-linear least squares methods in Python. Available from: https://ipython-books.github.io/93-fitting-a-function-to-data-with-nonlinear-least-squares/ [accessed 13 Sept. 2019]

4. Newville, Matthew. Non-Linear Least-Squares Minimization and Curve-Fitting for Python. Release 0.9.9+0.gb6f5789.dirty. Available from: https://buildmedia.readthedocs.org/media/pdf/lmfit-py/0.9.9/lmfit-py.pdf [accessed 13 Sept. 2019]

Authors

Aparna Komarla
NSG DC NAND Operations
Program Analyst Intern
aparna.komarla@intel.com

Michael Roten
NSG DC NAND Operations
Technical Project Analyst
michael.g.roten@intel.com