Predicting and Prescribing Admissions Success: Holistic Regression and Constrained Policy Trees

Student Members

Bennett Hellman

Iggy Siegel

Professor

Dimitri Bertsimas

December 3, 2021

Contents

1 Introduction

1.1 Background

At the United States Military Academy (USMA), candidates for admissions must solicit recommendations from Math, English, Science, and Physical Education teachers to complete their application. Since USMA routinely receives thousands of applications each year, and each applicant must submit four letters of recommendation, USMA ultimately receives tens of thousands of letters of recommendation each year. Unfortunately, admissions personnel rarely evaluate individual letters of recommendation at a thorough level because the volume of letters received prohibits objective analysis.

Currently, admissions officers at USMA combine all four recommendations into a single a score. Interestingly, this score does not explicitly target the contents of the recommendation letters. Rather, the score is a weighted sum of teacher responses to questions about the applicants such as: "This applicant has demonstrated an ability to work effectively with others toward group goals." The letter of recommendation writers answer twelve of these questions about the candidates, specifying the degree to which they agree with each statement on a scale from 1 (Agree strongly) to 5 (Disagree strongly). The total recommendation score is calculated by taking the arithmetic mean of the survey responses to the four recommendation letters, and this metric accounts for ten percent of a candidate's total application score. This implies USMA almost entirely relies on survey questions to assess recommendation letters because of the sheer amount of time it would take to thoroughly read every single letter. With this in mind, we would like to find a better way of assessing recommendation letters through machine learning techniques.

The first part of this project is predictive. Given an applicant's letters of recommendation, what is his or her predicted future performance at USMA? The second part of this project is prescriptive. Given an applicant's predicted future performance, what is the optimal admissions decision USMA should make concerning this applicant?

1.2 Data

The data set is a database of USMA students from the years 2016-2022. The database contains information on all 8,251 students admitted during these seven years. It not only includes information from the admissions committee but also detailed performance variables after the students arrived at USMA. The admissions variables consist of information such as standardized test scores, high school class rank, and high school GPA. The performance variables consist of information such as college GPA, class rank, and USMA-specific outcome variables such as whether the student was sent to a one-year college preparatory school. Finally, the database also includes letters of recommendation from Math, English, Science, and Physical Education teachers. In totality, the database consists of pre-admissions variables, post-admissions variables, and recommendation letters for 8,251 admitted students from the years 2016-2022. The following table shows some summary statistics of the students:

Characteristic	Data	
Gender, No (%)		
Male	6534	(79.2)
Female	1717	(20.8)
Ethnicity, No(%)		
Asian	649	(7.9)
Black	1093	(13.2)
Hispanic	760	(9.2)
Native American	81	(1.0)
White	5498	(66.6)
Other	170	(2.1)
Prior Military Service, No (%)		
Yes	517	(6.3)
No	7734	(93.7)
Graduation rate (2016-2017), No (%)		
Yes	1901	(85.9)
No	311	(14.1)
Standardized tests, mean (SD)		
SAT Math	640.9	(76.2)
SAT Verbal	623.7	(80.9)
ACT Composite	28.6	(3.7)
Graduation GPA, mean (SD)	3.12	(.20)

Table 1: Summary of data

1.3 Natural Language Processing

With a full data set, we began looking for systematic indicators in recommendation letters that suggest an applicant would succeed at USMA. One of the first ideas was to simply examine the raw number of words a recommender wrote about an applicant. This idea prompted us to compute four metrics about the letters: the total word count of all four letters, the word count of the shortest letter, the word count of the longest letter, and the average word count across all four letters.

After computing these simple word count metrics, we examined sentiment analysis metrics, which automates the interpretation and classification of emotions from text. Based on previous research, we decided to use the "afinn" lexicon to calculate total sentiment as the number of positive words minus the number of negative words in a recommendation letter. Additionally, we created two more sentiment metrics–PPS (proportion positive sentiment) is the number of words with positive sentiment divided by the total number of words, and PNS (proportion negative sentiment) is the number of words with negative sentiment divided by the total number of words.

At this point, we had created a handful of variables–such as word count and sentiment–but we wanted to create even more explanatory variables for the recommendation letters. Examining previous research in this area led to a computer program called Linguistic Inquiry and Word Count (LIWC), which contains over eighty different word dictionaries to capture nearly all types of words people use in everyday language [Pennebaker]. For example, the program contains dictionaries for pronouns, prepositions, verbs, cognitive processes (cognitive words such as "know" and "cause"), perceptual processes (perceptual words such as "look" and "heard"), and comparisons (comparison words such as "bigger" and "best"). In the end, after combining each applicant's letters of recommendation into a single chunk of text and running this through LIWC, we had created ninety-three additional predictor variable about the recommendation letters. However, some of the LIWC output seemed trivial. For example, punctuation variables such as the percentage of colons or parenthesis did not seem relevant. Additionally, we had concerns about variables with low base rates (less

than .5%) because a small number of recommendation letters could unduly impact the model. As a result, we removed the punctuation and low base rate variables in the LIWC output, trimming down to eighty-seven total language variables between the word count, sentiment, and LIWC output. As a final data preparation step, we standardized every remaining variable in the data set to zero mean and unit variance.

2 Predictive Models

2.1 Methodology

Out of a pool of eighty-seven potential explanatory variables, we wanted to pick out the variables that best predicted academic GPA. Equally important, we wanted the final model to be interpretable with a small number of explanatory variables that "point" in the direction expected. Our first attempt at a model began by examining correlations between variables. Although most of the predictors were slightly correlated with GPA, only ten were strongly correlated–meaning the absolute value of the correlation is greater than 0.15. The following graph shows the correlations of these predictors:

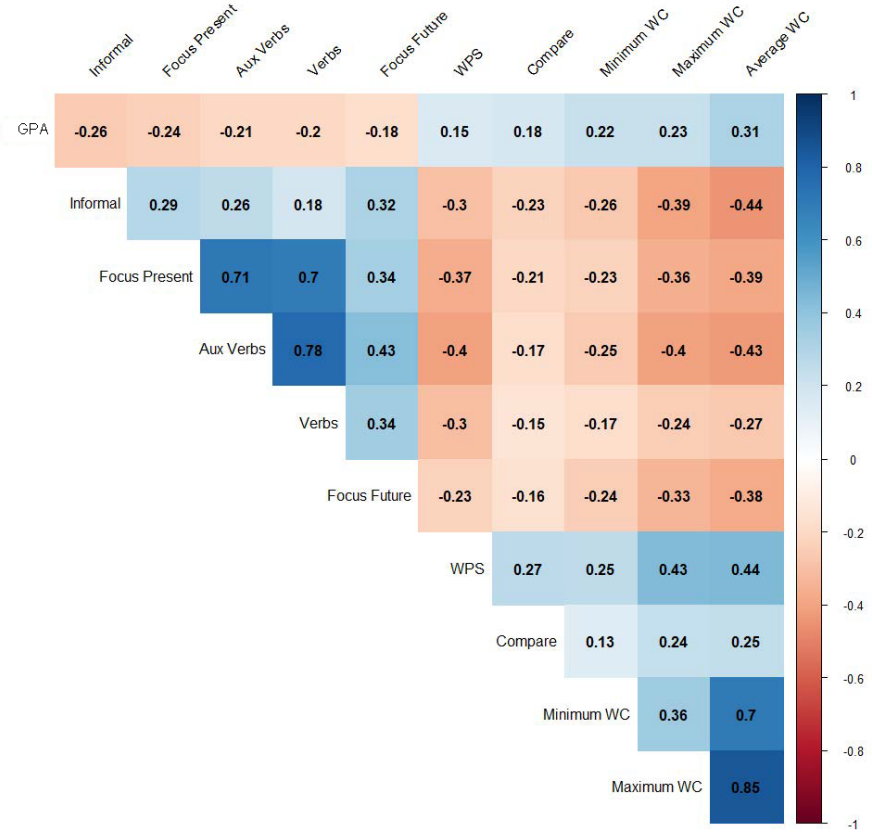

Figure 1: Correlation plot

Looking at the top row of the correlation plot, we see several variables are highly correlated with grades. The more important part of the story is that many of the variables are highly correlated with each other, meaning they are essentially the same concept. For example, the minimum, maximum, and average word counts are extremely highly correlated with one another, as are verbs, auxiliary verbs, and present focus. There is a sense only a few factors are driving all the variation in grades. After some experimentation, we found that just a five-term model not only captures most of the factors in the correlation plot but also does a good job predicting GPA. Furthermore, the coefficients for each of the terms points in the direction we

would expect based off the correlations, meaning this model is interpretable. In the end, after splitting the data into an 80% training set and 20% testing set, we fed these five factors into the Julia glmnetcv package to optimally tune Lasso robustness on the training set and evaluated performance on the testing set [Dunn].

Besides the sheer amount of time expended, the major issue with the previous model is that it takes a very qualitative approach to the problem. Given the power of MIO solvers, we decided to make a second model that implements the holistic regression approach. More specifically, given initial data $X \in R^{n \times p}$ and $y \in R^n$ we replaced each feature X_j with the following transformations:

$$T_j = \{\tilde{X}_{4(j-1)+1} := X_J, \tilde{X}_{4(j-1)+2} := X_j^2, \tilde{X}_{4(j-1)+3} := \sqrt{|X_j|}, \tilde{X}_{4(j-1)+4} := \log(|X_j|)\}$$

Then we performed the following holistic regression problem:

$$\begin{aligned} \min \quad & \frac{1}{2}||y - \tilde{X}\beta - \beta_0|| + \Gamma||\beta||_1 \\ \text{s.t.} \quad & -Mz_i \leq \beta_i \leq Mz_i, \quad i = 1, ..., 4 \cdot p \\ & \sum_{i=1}^{p} z_i \leq k \\ & \sum_{i:\tilde{X}_i \in T_j} z_i \leq 1, \quad j = 1, ..., p \\ & z_i + z_j \leq 1, \quad \forall i, j \in HC(\tilde{X}) \\ & z_i \in \{0, 1\}, \quad i = 1, ..., 4 \cdot p \end{aligned} \tag{1}$$

Where the term $\Gamma||\beta||_1$ models lasso robustness, the constraint k models sparsity, the set T_j allows at most one version of each transformed variable to appear in the model, and the set $HC(\tilde{X})$ controls pairwise collinearity. For hyperparameter tuning, we set the maximum pairwise collinearity parameter $\rho = 0.7$ and the sparsity parameter $k = 5$. Then, we split the training data into a 70% training set and 30% validation set to cross-validate the robustness parameter Γ across a set of values ranging from 0.001 to 2.0. The model that performed best on the validation set, according to the R^2 criterion, was selected as the final model.

2.2 Results

After building each model on the training set, we evaluated performance on the testing set according to the R^2 criterion. The following table demonstrates the results of each model:

Model	**Testing R^2**	**Training R^2**	**Sparsity**	**Optimal Γ**
Baseline Model	0.1779	0.1927	5	0.0011
Holistic Model	0.1893	0.2021	5	1.0

Table 2: Predictive model results

As seen in Table 2, the holistic model uses the same number of coefficients as the original model, but performs better on both the training and testing sets. Although a 0.01 increase in R^2 on the testing set may not seem significant, it is a 6.4% model performance increase given the low-signal environment of the problem. The holistic approach also has time advantages. Although running and cross-validating the MIO problem on a larger data set ($n = 8,251, p = 87$) took several hours to complete, this was a fraction of the time compared to the original approach of examining correlations, performing forward/backward/best subsets selection, and experimenting with combinations of variables until we found something that made sense.

On a separate note, we found that the greatest improvements to the models occurred through imputation techniques. Even though the data set was very "clean," we noticed that some applicants had all zero values

for the natural language processing variables. This corresponds to applicants whose teachers simply filled out the surveys at the end of the letter of recommendation without writing any words. By using optimal imputation techniques, the performance of the holistic regression model increased as follows:

Model	**Testing R^2**	**Training R^2**	**Sparsity**	**Optimal Γ**
Holistic Model	0.1893	0.2021	5	1.0
Holistic Model with Imputation	0.2003	0.2022	5	1.0

Table 3: Predictive model results with imputation

As seen in Table 3, optimal imputation resulted in a 5.8% downstream model performance increase compared to the original holistic regression approach, and a 12.6% downstream model performance increase compared to the baseline model.

In the end, we used automated text analysis to create a model that assesses thousands of recommendation letters in a matter of seconds. The baseline model has an R^2 value of .1779 while the holistic approach with optimal imputation has an R^2 value of .2003. Given that traditional college admissions metrics such as high school GPA only predict college GPA with an R^2 value of .1826 for this data set, it certainly appears as if simple text analysis holds value for making more objectively informed admissions decisions. When trying to find an interpretable and high-performing technique, holistic regression combined with optimal imputation performs very well.

3 Prescriptive Models

3.1 Background

So far we have been focusing on predicting success at USMA through letters of recommendation. Letters of recommendation seem to hold some signal in predicting GPA, but we could make a much stronger model by incorporating other variables in the data set such as standardized test scores, high school GPA, high school class rank, and a set of binary variables such as whether an applicant was a recruited athlete or previously served in the military. Although creating a strong predictive model might be useful for USMA, at this point we would like to take a broader view of the issue.

Every year, USMA sends approximately 20% of its accepted applicants to a one-year preparatory school. The question naturally arises: which subpopulation of accepted applicants should we send to the preparatory school? The following optimal classification tree demonstrates who has entered the preparatory school in the past:

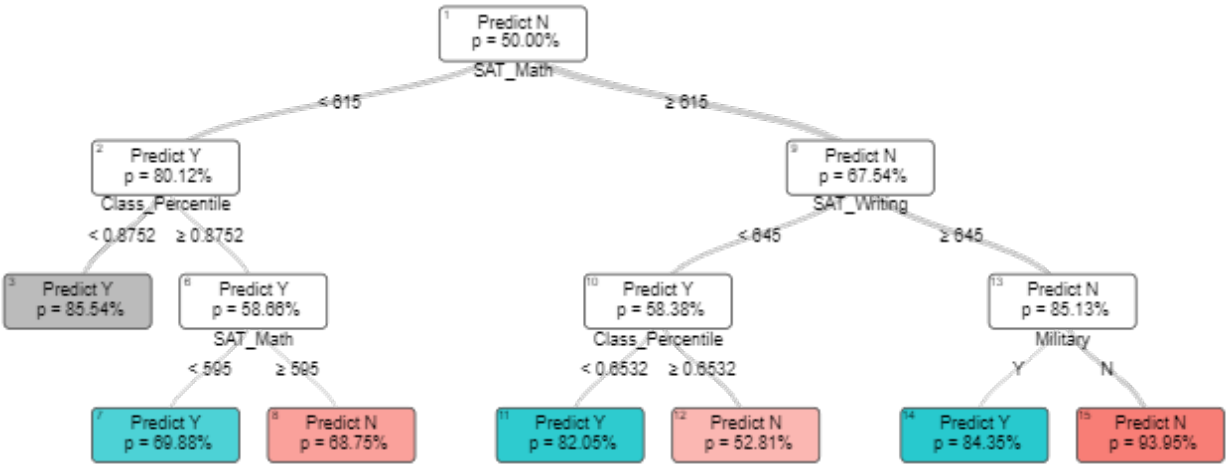

Figure 2: Current USMA preparatory school policy

As seen in Figure 2, the current policy makes intuitive sense. In general, SAT Math is the most important factor, and those with a combination of low high school GPA and low standardized test scores typically get referred to preparatory school. Additionally, applicants with high standardized test scores but who previously served in the military also typically get referred to preparatory school. The rationale is that these applicants have not been exposed to an academic setting for several years and could benefit from a year of preparation. Even though the current policy makes sense, it does not necessarily mean that the optimal subpopulation of applicants is sent to the preparatory school to maximize overall academic performance at USMA. In this section, we would like to apply policy tree techniques to find this optimal subpopulation.

3.2 Optimal Policy Trees

Under a strictly predictive framework, the admissions committee would calculate projected college GPAs for all the applicants and then send the lowest performing 20% to the preparatory school. Given that we have such a large data set, it would be better to apply a prescriptive framework to decide which applicants should attend the preparatory school. The major issue with this approach is that we lack counterfactuals–we do not know what would have happened if we did not send someone to the preparatory school and vice-versa. To compute these counterfactuals and determine the optimal policy, we created a causal inference reward matrix through the doubly robust method and created an IAI optimal policy tree based off this reward matrix. As in the previous section, before creating this optimal policy tree we performed optimal imputation:

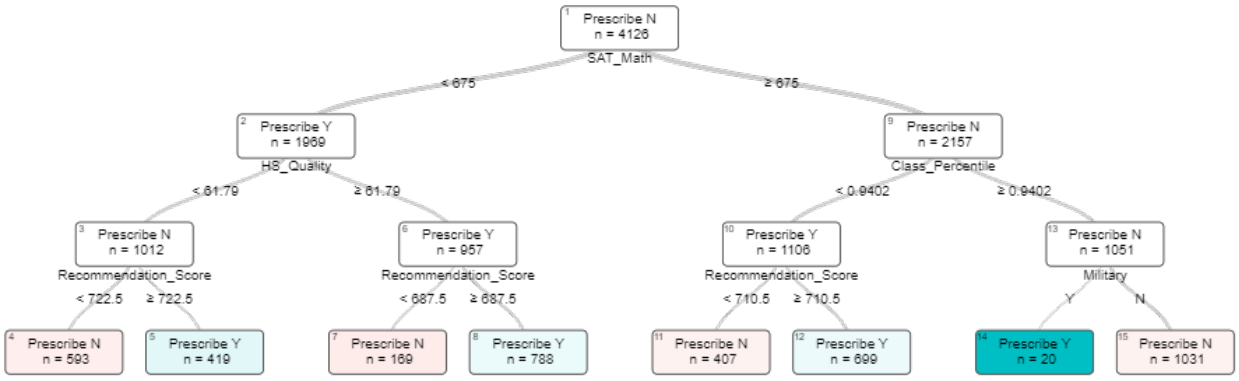

Figure 3: Optimal policy tree for USMA preparatory school decision

As seen in Figure 3, the optimal policy tree sends a similar but slightly different group of applicants to the preparatory school. Again, SAT math is the most important factor, and those who previously served in the military typically get referred to the preparatory school. Interestingly, those with a higher letter of recommendation score also typically get referred to the preparatory school. It is important to note that this recommendation score is based off the old USMA survey scoring system. This provides further evidence that the old system does not do a good job of differentiating success at USMA; in fact, one might argue that applicants with a higher recommendation score actually perform worse at USMA. In the end, by assigning applicants to the preparatory school according to this optimal policy tree, we predict a small increase of 0.019 overall GPA at USMA.

3.3 Constrained Optimal Policy Trees

The major issue with the previous policy tree is that there is no constraint on the number of applicants we send to the preparatory school–the model recommends we send 47.6% of applicants on a one-year hiatus. In reality, the preparatory school only has room for a maximum 20% of applicants each year. Unfortunately, the current IAI software package does not directly support a way to limit the number of people allocated to a given treatment, but it is something that can be added through a post-hoc processing of the tree. In this section, we extend IAI's optimal policy tree to incorporate this constraint.

Given an optimal policy tree and an associated reward matrix, the goal is to set up a simple MIO problem with variables to assign treatments to applicants with the objective to maximize the reward of the assigned treatments. Then we can add a constraint that only 20% of treatments total can be assigned, and also a constraint that all patients in the same leaf of the tree must get the same treatment. When solving this problem, we effectively have a new tree that applies a single treatment in each leaf as before, but with the constraint that only 20% of treatments are applied across the whole tree. The following formula explicitly defines the MIO problem:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} x_i \cdot r_{t_i} + (1 - x_i) \cdot r_{nt_i} \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_i \leq k \cdot n \\
& \sum_{i \in L_j} x_i \geq l_j \cdot |L_j|, \quad \forall j \in \text{Tree} \\
& x_i \leq l_j, \quad \forall i \in L_j \\
& x_i, l_j \in \{0, 1\} \quad \forall i, j \\
\text{where} \quad & r_{t_i} = \text{Reward of treatment}, \quad \forall i \\
& r_{nt_i} = \text{Reward of no treatment}, \quad \forall i \\
& k = \text{Proportion of treatments available} \\
& |L_j| = \text{Size of Leaf } j, \quad \forall j \in \text{Tree} \\
& x_i = \begin{cases} 1 & \text{Applicant } i \text{ attends preparatory school} \\ 0 & \text{Otherwise} \end{cases} \quad \forall i \in [n] \\
& l_j = \begin{cases} 1 & \text{Leaf } j \text{ selected in policy tree} \\ 0 & \text{Otherwise} \end{cases} \quad \forall j \in \text{Tree}
\end{aligned}
\tag{2}
$$

Given that Equation 2 contains multiple binary variables and non-linearity in the objective function, we were worried about the tractability of the formulation. Practically speaking, Gurobi solved this problem in less than a second. In the end, the constrained policy tree recommends we send the following applicants to preparatory school:

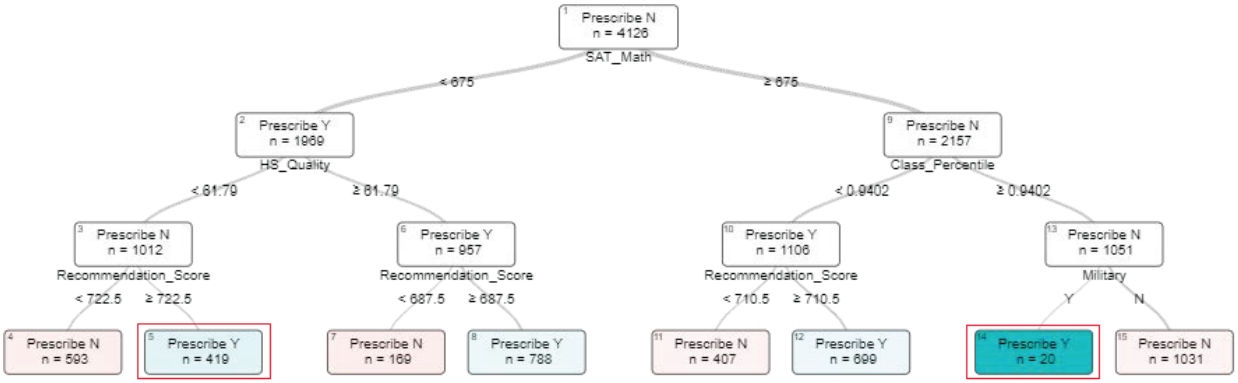

Figure 4: Optimal constrained policy tree for USMA preparatory school decision

In the constrained policy tree shown in Figure 4, only 10.4% of applicants are sent to preparatory school, and we predict a slight increase of .0037 overall GPA at USMA.

4 Conclusion

Overall, we provide improvement to the USMA admissions process through predicted success and prescribing preparatory school. Through holistic regression, we both improve predictive performance as well as provide interpretable solutions about which sparse number of attributes contribute to a successful applicant. Additionally, we created a policy tree to suggest which applicants should attend preparatory school before USMA to increase overall GPA (albeit very slightly).

Before closing, we would like to provide two final remarks. First, our MIO formulation for constrained policy trees is an improvement on the baseline IAI package, but it is not complete. Under the current framework, we only send 10.4% of applicants to preparatory school even though we have room for 20% of applicants. A better formulation would apply the remaining treatments to the policy tree leaf that best benefits, even if this would violate the MIO constraint by allowing some of the individuals in that leaf to not receive the same treatment.

Second, this project uncovered an interesting case-study about the role of bias in data. Although this data set originally contained hundreds of variables, we decided to get rid of many variables for equitability purposes. For example, Table 1 displays several summary statistics about the applicants, but we never included gender or ethnicity variables in the predictive models. Nevertheless, proxies of these variables tried to creep into the models through indirect measures. One of the natural language processing variables that was very significant is LIWC's "female" variable, which calculates the probability a piece of unstructured text was written by a female. Although we ultimately removed this variable despite is high predictive ability, this project demonstrates the importance of removing biases from data, especially in high-impact scenarios such as admissions decisions.

Appendix A - Member Contributions

This project was written by Bennett Hellman (MBAn '22) and Iggy Siegel (ORC '23). Given that both of us are graduates of USMA, we naturally have a vested interest in the school and looked for ways to use optimization and machine learning techniques to improve decision-making in the world directly around us. Although much of the work was done together, Bennett Hellman primarily focused on data cleaning, data imputation, initial unconstrained policy tree coding and project presentations. Iggy Siegel primarily focused on the holistic regression and policy tree coding, as well as the MIO extension for constrained policy trees.

References

Dunn, Jack. GLMNet.jl. https://github.com/JuliaStats/GLMNet.jl. Accessed 2 Dec. 2021.

Pennebaker, James. LIWC. http://liwc.wpengine.com/. Accessed 2 Dec. 2021.