The Effects of Constituent Characteristics on ESP Matching

Ethan Lau
University of California, San Diego
Alumni, Career, Annual Giving, Pipeline Development, and International Department
Supervisor Charlie King
August 25th, 2024

This page intentionally left blank.

TABLE OF CONTENTS

1—ABSTRACT....4
2—DATA ANALYSIS & PROCESSING....4
2.1 DATA SELECTION & CLEANUP....4
2.2 VARIABLE ANALYSIS....6
2.2—A EVENT DATE....6
2.2—B ADDRESS....7
2.2—C CATEGORICAL VARIABLES....8
ATTENDANCE TYPE....8
REGISTRATION TYPE....9
2.2—D BINARY VARIABLES....11
TRENDS & SIGNIFICANCE....11
MODERATE VARIABLES....12
BIASES & OUTLIERS....12
2.3 MULTICOLLINEARITY & BIAS HANDLING....13
3—MODEL RESULTS....14
3.1 OLS MODEL....15
3.2 LASSO & RIDGE....16
3.3 STEPWISE....18
3.4 HIGH DEGREE SPLINE....19
Model 1: High Degree Spline on ev_date, Event Date....21
Model 2: High Degree Spline on address, Number of address fields filled....21
Model 3: High Degree Spline on ev_date and address....21
4—CONCLUSIONS....22
4.1 FINAL MODEL RESULTS....23
4.2 EVENT ANALYSIS....26
4.3 FINAL CONCLUSIONS....29
5—WORKS REFERENCED....30
6—APPENDIX....31
APPENDIX A—FINAL MODEL OUTPUT....31
APPENDIX B—REGISTRATION TYPE STATISTICS....38
APPENDIX C—BINARY VARIABLE SUMMARY QUANTITIES....39
APPENDIX D—TRAINING DATA CODE....40
APPENDIX E—NAIVE & TEST DATA TRAINING CODE....48

Distribution: *Privately Accessible.*
***Not Publishable for Any Purpose.**

1—ABSTRACT

The ability to accurately match constituent information to existing records in a CRM system is critical for maintaining data integrity and driving engagement efforts. At the University of California, San Diego Alumni Relations Department, the challenge is pronounced due to the non-uniform nature of event registration forms, which vary across events and create inconsistencies in the collection of constituent data. Open-ended fields and inconsistent formats make it difficult to standardize and clean the data, resulting in avoidable gaps or inaccuracies when trying to match registrants to their existing IDs in the CRM database known as the Engagement and Stewardship Platform or ESP.

This paper focuses on analyzing a dataset consisting of one-hundred-thirty-six (136) alumni relations events, identified with the prefix ACE, which span from November 13, 2014, to July 27, 2024. This paper aims to determine which constituent information most significantly influences the ability to match registrants to an existing ID.

2—DATA ANALYSIS & PROCESSING

2.1 DATA SELECTION & CLEANUP

The dataset—derived from the aforementioned alumni relations events—yielded a total of sixty-four-thousand-eight-hundred-fifty-five (64,855) rows of registrant information. The selection of columns for analysis was based on two criteria: the presence of consistent column labels across multiple events and the number of non-blank entries within each column. This approach ensured that only relevant and uniformly applicable columns were included in the dataset. Additionally, the *source_id* and *id_extracted* columns were removed as they are possible confounding variables of the *id_found*.

To maintain data quality and integrity, several cleanup procedures were implemented. Initially, cells containing placeholder values such as n/a, na, null, dnc, and "–" were removed. Further manual data cleaning was conducted to address specific anomalies, including the removal of cells that contained only commas or nonsensical entries, such as "not interested" in fields where numerical values like graduation year were expected. This meticulous process involved manually adjusting fewer than a thousand (1,000) cells out of nearly two-million-seven-hundred-thousand (2,700,000) total cells.

Rows with null event names were excluded from the dataset to ensure that only entries with valid event identifiers were retained. The final step in the data preparation involved filtering the dataset to include only those rows marked as reviewed and removing those columns where the variance was less than a threshold of 0.0005 to reduce overfitting. These rigorous data cleanup and selection processes were crucial for achieving better accuracy and reliability in the subsequent analysis.

Category	Variable	Description	Type
IDs	id_found	If ESP ID was found	Dependent
	source_id	If provided source ID was used	Binary
	id_extracted	If ID was queried based on contacts	Binary
Event Attributes	event_id	Event Names	Dummy
	ev_date	Date of the Event	Continuous
Review Status	id_stat	If a new or existing constituent, or special circumstance	Filter Variable
	reviewed	If manually reviewed	Binary
Constituent Characteristics	first_name	First name provided	Binary
	last_name	Last name provided	Binary
	org_name	Organization name provided	Binary
	name_title	Mr., Ms., Mrs., Dr., or Sir.	Binary
	job_title	Job title provided	Binary
	reg	Registration type	Dummy
	academ_divis	Academic division provided	Binary
	college_school	College or school provided	Binary
	grad_yr	Graduation Year provided	Binary
	address	Number of address details provided	Continuous
Contacts	email	Email provided	Binary
	phone	Phone provided	Binary
Guest Status	is_invitee	If an event invite	Binary
	is_guest	If an event guest	Binary
	guest_of	If guest indicated their invitee	Binary
	invite_accepted	If invitation accepted	Binary
	invite_rejected	If invitation rejected	Binary
Participation	reg_date	Registration date	Binary
	yes_partic	If participated	Binary
	no_partic	If did not participated	Binary
	cancel_date	If canceled registration	Binary
	modif_date	If registration details were modified	Binary
	action_date	If something was done to registration	Binary
	atten_type	Way in which event was attended	Dummy

Table 1: Dataset Variables

2.2 VARIABLE ANALYSIS

In this section, we embark on a crucial phase of our analyses journey by thoroughly examining the dataset and conducting preliminary analysis. Our focal point lies in understanding the dynamics surrounding the selected dependent variable—if the constituent was matched or not (*id_found*). To unravel the interplay between the matching process and its determinants, we exercise discretion in selecting variables that hold substantive relevance.

Density histograms and bar charts are employed to visualize the distributions of continuous and categorical variables respectively, revealing the frequency and spread of data points. For binary variables, a radial binary variable plot is used to illustrate the proposed relationship. This chart highlights how different binary attributes interact and their impact on matching success. The insights gained from these analyses are intended to inform data collection and predictive modeling efforts, offering a foundation for more effective strategies in managing and utilizing registrant data.

2.2—A EVENT DATE

The variable *ev_date* encompasses events ranging from 2014 to 2024, with a notable concentration of events occurring between 2021 and 2024. It is important to note that this temporal distribution reflects the structure of the dataset rather than an actual increase in event frequency during these recent years.

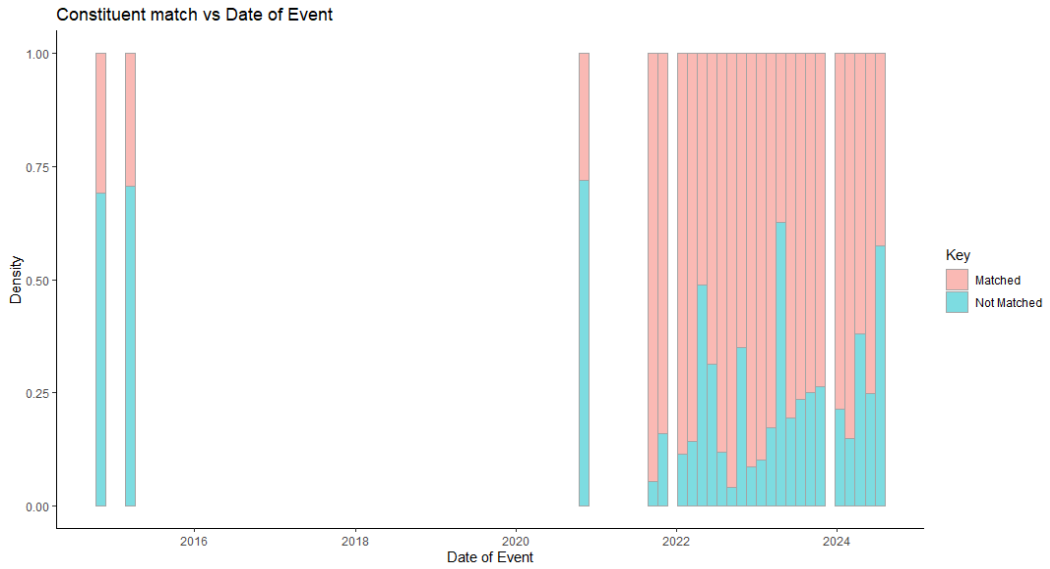

Figure 1: Continuous variable—matched density binned by event date.

Preliminary analysis indicates that events from the past four years exhibit a higher initial rate of successful matches. This trend may be attributed to the review of events held before 2022 being

conducted in 2023, resulting in the creation of numerous new-constituent records in the CRM system. Consequently, the enhanced match rate for these earlier events is likely due to that increased availability of updated constituent information.

Conversely, the number of non-matches has risen for events closer to the present day. This rise can be attributed to the influx of new students and families participating in more recent events, leading to a higher proportion of unverified or incomplete records. As a result, the data suggests that while earlier events had higher matching success due to updated records, recent events face challenges related to the integration of newly added constituents. Controlling for the variable *ev_date* in subsequent analyses will mitigate the observed bias, permitting adjustment for the differential impact of retrospective updates and the influx of new constituents. This adjustment will enhance accuracy of the assessment of the true relationships between constituent information and matching success.

2.2—B ADDRESS

The variable *address* reflects the number of address fields provided, including address line, city, state/province, country/region, and zip code. The values in principal range from zero (0) to five (5), where a value of five (5) indicates the provision of all address fields. However, the maximum number of address fields in the dataset is three (3). An example of three (3) fields present would be if a constituent provided address, state, and zip code or state, zip code, and city.

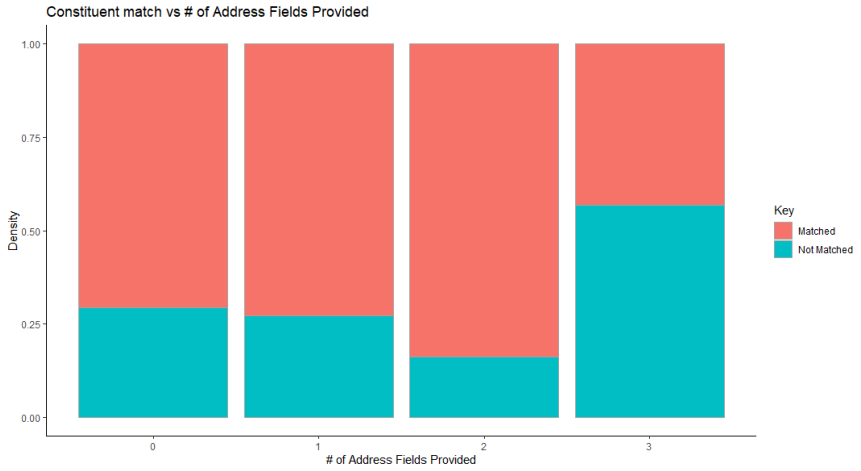

Figure 2: Continuous variable—match density for number of address fields.

Empirical analysis reveals a slight relationship between the number of address fields provided and the success rate of matching to existing IDs in the CRM system. Registrants with no address fields exhibit a roughly 75% match rate, while those with a single address field show a marginally lower match rate. The match rate improves for those providing two address fields, indicating that more comprehensive address

information generally facilitates better matching. However, the match rate for registrants providing three address fields declines significantly. This anomaly may be attributed to several factors, including potential issues with address formatting or the presence of outdated or erroneous data. Additionally, within the current standard operating procedure (SOP) for manual matching, address is utilized as a secondary matcher. Notably, unless a constituent updates their address, the initial address provided during registration is retained as their official address—such as a student's on-campus address, which remains the registered address unless explicitly updated. Therefore, further investigation is needed to explore how address information interacts with other variables and to assess its impact on the overall effectiveness of the matching process. It may also be necessary to remove said variable dependent on the coefficient bias.

2.2—C CATEGORICAL VARIABLES

There are two primary categorical variables: *atten_type* (for attendance type), which reflects the mode of participation in events, and *reg* (for registration type), which categorizes registrants by their affiliation with the university, such as student, alumni, or staff. Given that university affiliation and engagement improves identification of a constituent, it is probable both variables impact match success.

ATTENDANCE TYPE

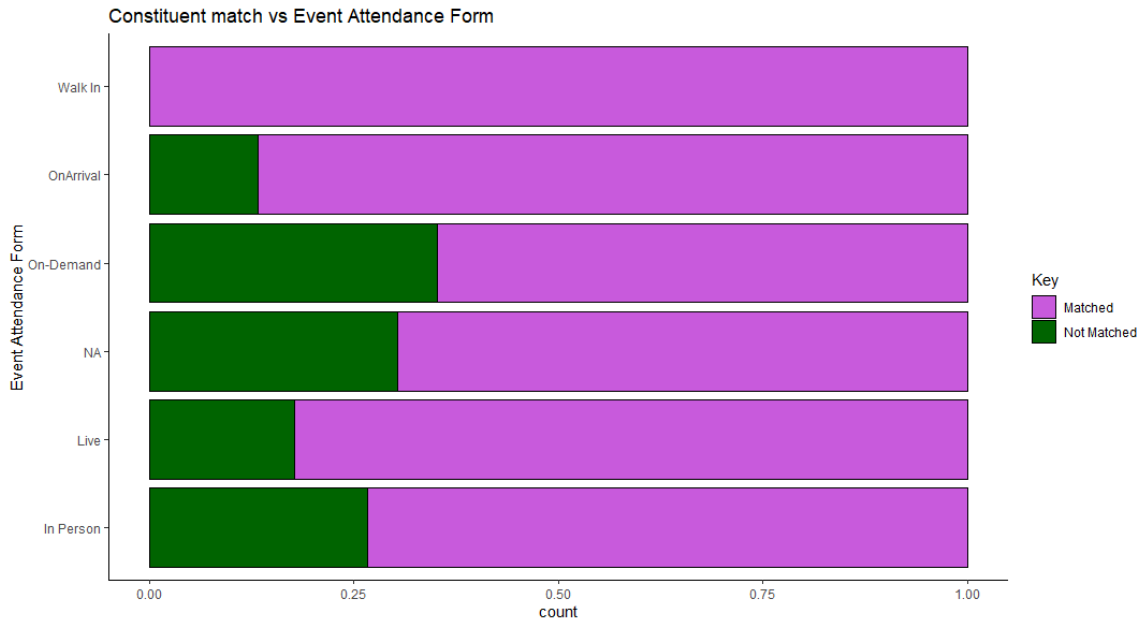

Figure 3: Categorical variable—match density for different attendance types.

Analysis reveals notable variations in matching success across different attendance. Registrants classified as "walk-in" demonstrate a 100% match rate, indicating complete success in aligning these attendees with existing IDs. In comparison, those marked as "on-arrival" achieve a roughly 85% match rate, reflecting a

relatively high level of successful matches, but also highlighting a notable proportion of unmatched entries. Registrants categorized under “on-demand” exhibit a roughly 60% match rate, suggesting a more considerable challenge in achieving successful matches within this group. The “NA” category, where attendance information is absent or ambiguous, results in a roughly 70% match rate, implying that incomplete data may hinder matching accuracy. Registrants identified as “live” and “in person” show match rates of roughly 80% and 75%, respectively, indicating intermediate success in matching.

These discrepancies in match rates among different attendance types suggest that the method of event participation may significantly influence the effectiveness of matching registrants to existing IDs. However, given over 50% of the dataset are nulls for attendance type, it’s difficult to draw a conclusion. Those registrants recorded as “on-demand” or “walk-in” also have a sample size of less than a hundred (100) out of the sixty-five-thousand (65,000) participants. This may call into question the efficacy of the relevance of those particular categories.

REGISTRATION TYPE

The variable for registration type classifies registrants by their affiliation or status, such as student, alumni, or staff. The varying match rates observed across registration types can be logically attributed to the nature of each affiliation.

For instance, “Student” registrants, with a large sample of roughly eighteen-thousand (18,000), have a match rate of approximately 75%. This high rate is expected because students typically possess unique university identifiers—such as student IDs—which are directly linked to the institution’s records. Their ongoing interactions with the university contribute to a high likelihood of successful matches.

Similarly, “Alumni” registrants, numbering over twelve-thousand (12,000), show a match rate of roughly 70%. Alumni, having been previously affiliated with the institution, often retain their unique identifiers, which facilitates their match. The same reasoning applies to “Staff” and “Staff/Faculty Alumni,” with match rates of roughly 72% and 74%, respectively. Staff members and alumni with staff affiliations are likely to have well-documented identifiers, making them more easily matchable.

In contrast, “Community Member” registrants, with slightly over four-thousand (4,000) entries, show a lower match rate of roughly 65%. Community members generally lack direct affiliations with the university, such as student or staff IDs, leading to less effective matching. This group’s lower match rate reflects the absence of systematic identifiers tied to the university, making it harder to match their records. Similarly, categories such as “Minor” and “VIP” demonstrate how the presence or absence of university identifiers affects match rates. “Minor” registrants have a match rate of roughly 60%, reflecting potential

challenges in aligning their records due to less frequent engagement. Conversely, "VIP" registrants, though numbering fewer than a hundred (100), have a higher match rate of roughly 75%, likely due to their notable status and associated records.

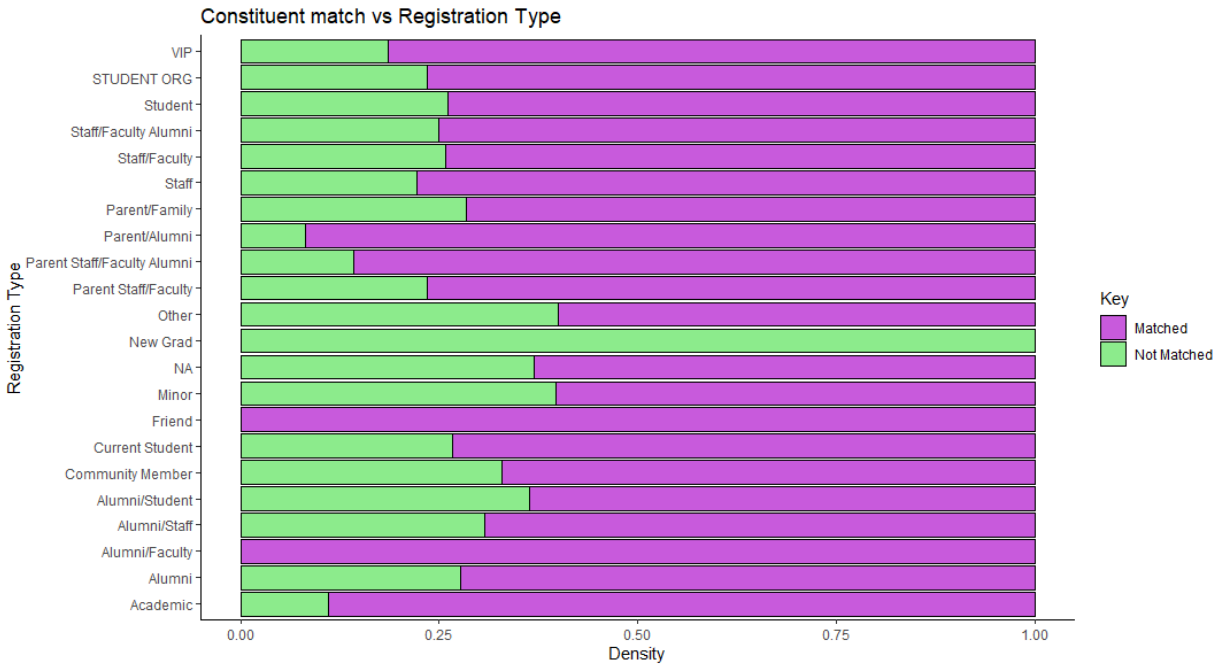

Figure 4: Categorical variable—match density for different registration types.

The analysis of registration types reveal how the nature of registrant affiliations logically influences matching success. Categories with strong ties to the university, such as students, alumni, and staff, generally exhibit higher match rates due to the presence of unique university identifiers. In contrast, categories with weaker ties, such as community members, show lower match rates. Understanding these associations helps explain the observed variations in matching success and highlights the importance of affiliation type in the CRM matching process.

However, it is important to note categories with very small sample sizes exhibit pronounced variability in match rates, which can potentially bias the overall analysis. The aforementioned "New Grad," "Parent Staff/Faculty Alumni," and "Friend" show high variability in match rates due to their limited sample size. The small number of registrants in these categories can lead to skewed results and potentially misleading conclusions about their matching effectiveness. The pronounced variability in match rates for these small samples underscores the need for caution when interpreting results. Given the dataset's substantial size, categories with few registrants may not provide a reliable representation of matching success.

2.2—D BINARY VARIABLES

Binary variables take on values of either true or false, providing a straightforward mechanism to analyze the presence or absence of specific information. The results from these variables, visualized through a radial binary variable plot, offer key insights into trends, potential biases, and the significance of each variable in determining a successful match within the CRM system. The output allows us to compare the match densities between true and false values for each variable.

TRENDS & SIGNIFICANCE

Several variables exhibit notable trends when comparing match rates between their true and false values. For example, *source_id* shows a stark contrast, with a match density of 98.73% when true and only 25.31% when false. This significant discrepancy indicates that the presence of a *source_id* may be influential in facilitating successful matches as it directly ties to pre-existing identifiers within CVENT that were imported from ESP. However, due to that deterministic connection, it may also be a confounding variable.

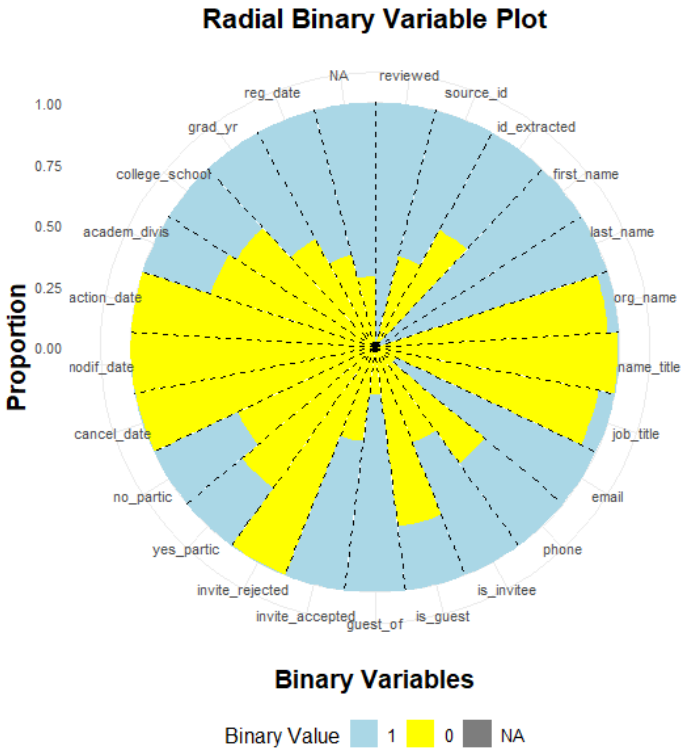

Figure 5: Binary radial variable plot for all binary variables.

Variables such as *first_name* and *last_name*, essential for constituent identification, also exhibit substantial differences between true and false values. When both are provided, the match density is

approximately 70%, while in their absence, the match rate drops to around 23%. This suggests that while the absence of a name does not preclude matching, it significantly reduces the likelihood of success, as missing core identification data challenges the ability to verify the constituent.

MODERATE VARIABLES

Some variables present a balanced distribution of match densities between true and false values. For instance, *email* has a match density of roughly 72% when true, but even when absent (false), the match density remains reasonably high at approximately 46%. This indicates that while email is an important identifier, it is not as crucial as other variables where the presence or absence drastically alters the match outcome.

Similarly, *phone* shows a relatively balanced effect, with a 66.83% match density when true and 71.36% when false. This pattern suggests that phone numbers may act as a secondary or redundant identifier, where their absence does not significantly impede matching as long as other key identifiers are available. However, it is important to note that the standard operating procedure (SOP) for matching did not utilize phone matching significantly until March of 2024.

In consideration of the patterns from phone numbers and source IDs, it may be revealing to A) perform an analysis on those constituents not matched using *source_id* as a determinant to unveil trends in those constituents not automatically discoverable and B) to control for the change in standard operating procedure regarding phone matching.

BIASES & OUTLIERS

Certain variables exhibit unique behaviors that merit attention for potential biases. For instance, variables like *modif_date* and *action_date* show perfect match densities when true (100%), but their low sample counts (twenty-six (26) and two (2) respectively) raise questions about the reliability and significance of these results. The small sample sizes for true values in these cases could skew the match density upwards, as the limited data might not adequately represent the overall population.

Variables such as *invite_rejected* show a relatively low match density of 62.35% when true, compared to the average match rates for most other binary variables. This outcome might suggest that constituents who reject invitations are less likely to be accurately matched in the system. However, with a small sample size of less than two-hundred (200) for true values, this variable also suffers from potential bias due to limited data representation.

2.3 MULTICOLLINEARITY & BIAS HANDLING

Multicollinearity occurs when two or more independent variables are highly correlated, leading to inflated standard errors and unstable coefficient estimates, making it difficult to assess the true impact of individual predictors on the dependent variable. Given the dataset's inclusion of variables with sparse counts, proper handling of multicollinearity is crucial.

In this study, multicollinearity is handled by running an Ordinary Least Squares (OLS) regression to help identify which variables show high correlation with one another or with the dependent variable. Variables like *new_grad* and *action_date*, with minimal occurrences, are likely to exhibit near-collinearity with the dependent variable. In such cases, their influence on the model is negligible or even distorted, as these rare events do not provide enough variation to offer meaningful contributions to the analysis. The almost perfect alignment between these small-row-count variables and the dependent variable causes them to be flagged as collinear, leading to their removal.

By filtering out variables with high collinearity and those with minimal counts, the model avoids being skewed by the effects of rare observations. This process is supported by other studies on multicollinearity which highlight the pitfalls of including variables with extremely small sample sizes in multivariate models. When variables with low occurrence are near-collinear with the dependent variable, they introduce noise without offering substantial insight, risking inflated coefficients and overfitting.

Moreover, to validate the robustness of the final model after multicollinearity handling, bootstrapping will be employed in the subsequent phases of analyses. Bootstrapping, a resampling technique, will allow for the testing of statistical significance by generating numerous subsets of the dataset and recalculating estimates. This approach will help ensure that the model's predictions are not unduly influenced by specific observations—especially those in small-row-count categories—but rather reflect genuine patterns in the data. Bootstrapping also provides confidence intervals for the model's parameters, offering a rigorous assessment of its predictive power across the entire dataset.

3—MODEL RESULTS

Fourteen (14) regression models were devised to analyze the impact of the aforementioned predictors on match success. These models encompass various methodologies, including ordinary least squares (OLS), ridge regression, lasso regression, forward stepwise regression, backward stepwise regression, bi-directional stepwise regression, and high-degree spline regression applied to selected variables. The outcomes of these models are summarized in Table 2.

Model	Naive MSE	Train MSPE	Test MSPE
Linear Regression (OLS)	0.126748	0.126848	0.283853
Ridge	0.127999	0.128093	0.127485
Lasso	0.126776	0.126879	0.126348
Forward Stepwise	0.126769		0.283817
Backward Stepwise	0.126770		0.283895
Bi-Directional Stepwise	0.126770		0.283895
Ridge with Polynomial Features	0.127905	0.127956	0.127335
Lasso with Polynomial Features	0.126708	0.126767	0.126167
Forward Stepwise with Polynomial Features	0.126707		
Backward Stepwise with Polynomial Features	0.126709		
Bi-Directional Stepwise with Polynomial Features	0.126709		
High Degree Spline on ev_date	0.126732	0.126151	0.283819
High Degree Spline on address	0.126710	0.126821	0.283857
High Degree Spline on ev_date & address	0.126696	0.126129	0.283817

Table 2: Model Results

The performance of each model was evaluated using the Mean Squared Prediction Error (MSPE) on both training and testing datasets. These results were benchmarked against Naive Models, which did not employ k-fold cross-validation, thereby providing a foundational baseline for comparative analysis. Both lasso and ridge regression models demonstrated robust predictive capabilities, particularly in configurations where polynomial features were incorporated.

In contrast, models integrating high-degree splines or OLS behaviors displayed marked increases in MSPE during the testing phase, with a pronounced divergence between training and testing errors. This substantial increase in MSPE suggests overfitting, where model complexity enables it to conform excessively to the training data at the expense of generalizability to unseen data.

Subsequent sections will provide a comprehensive examination of each model's performance, exploring the contributions of specific variables to predictive power and the effects of model complexity on generalizability. The overarching aim remains the identification of the most parsimonious model that effectively predicts successful ID matches.

3.1 OLS MODEL

The Ordinary Least Squares (OLS) regression model serves as a baseline for comparison against more complex models in this analysis. OLS assumes a linear relationship between the independent variables and the dependent variable, aiming to minimize the sum of squared residuals, or the differences between observed and predicted values. OLS provides an understanding of how well the data fits a simple linear model without additional transformations or regularization techniques.

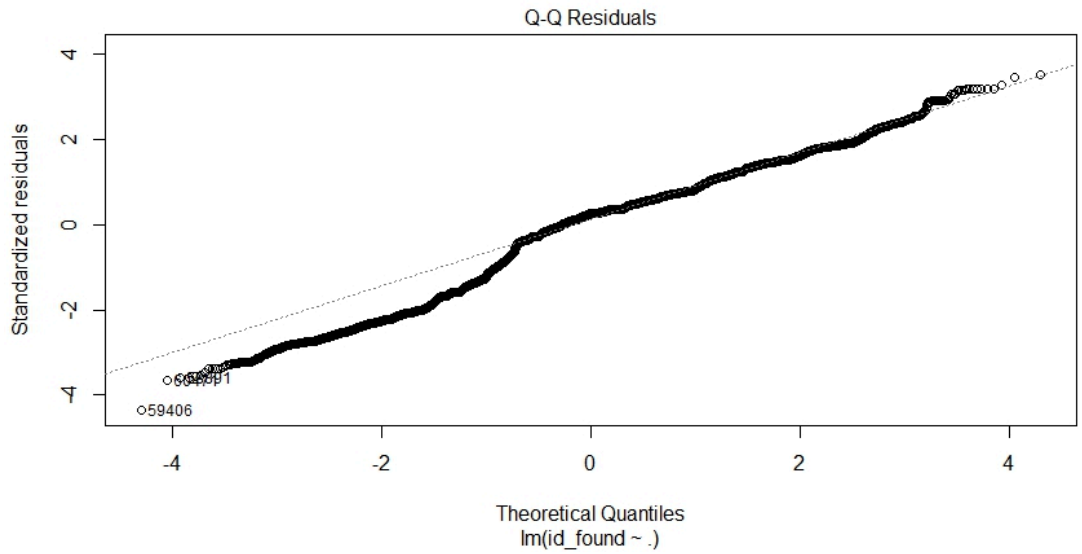

Figure 6: Cross-Validated OLS Quantile-Quantile Plot.

The OLS model achieves a relatively low MSE and training MSPE, indicating that it performs well on the training data. However, the substantial increase in testing MSPE (from 0.127 to 0.284) suggests that the model does not generalize well to unseen data. This significant disparity between training and testing performance indicates overfitting, where the model captures noise or irrelevant patterns in the training set rather than generalizable trends.

Given the simplicity of OLS, this increase in testing error is not unexpected, particularly when applied to a dataset of this size and complexity. The model's assumption of strictly linear relationships limits its ability to capture nonlinear interactions between variables, which likely accounts for its diminished performance on the test data. These results underscore the limitations of OLS in handling complex datasets, where linear assumptions can lead to suboptimal predictive accuracy. The evident overfitting reinforces the necessity of employing more sophisticated models that incorporate regularization techniques or non-linear transformations.

3.2 LASSO & RIDGE

Lasso and ridge regression are methodologies in the domains of machine learning and statistics, particularly used for their efficacy in regularization within regression models. These techniques prove especially invaluable when confronted with high-dimensional datasets or instances of multicollinearity among predictor variables. Both lasso and ridge regression strategies endeavor to mitigate overfitting tendencies by introducing a penalty mechanism governing the magnitude of coefficients. Ridge regression—also referred to as L2 regularization—integrates a penalty term into the linear regression objective function, precisely the sum of squared residuals (RSS). This penalty term is directly proportional to the square of the coefficients' magnitude. Notably, the regularization parameter λ assumes a pivotal role in dictating the extent of shrinkage applied to the coefficients. Higher values of λ engender more pronounced shrinkage, thereby steadily reducing the coefficients' magnitude towards zero. It is imperative to acknowledge that ridge regression, while effectuating proportional shrinkage across coefficients, never diminishes them to absolute zero, thereby preserving all predictors within the model. In contrast, lasso regression—known as L1 regularization—adopts a distinct approach by imposing a penalty term on the RSS, leveraging the absolute magnitude of coefficients. Unlike ridge regression, lasso regression exhibits a propensity for inducing sparsity within the model, effectively driving select coefficients to precisely zero. This intrinsic characteristic renders lasso regression particularly adept at feature selection, facilitating the automatic curation of a subset comprising the most pertinent predictors while concurrently attenuating the significance of others. Analogous to ridge regression, the regularization parameter λ plays a pivotal role in regulating the extent of shrinkage and sparsity engendered within the model architecture.

In the application of ridge or lasso regression with polynomial features, the regularization penalty extends to all coefficients, encompassing those corresponding to the polynomial terms. This strategic imposition of regularization serves the dual purpose of mitigating overfitting tendencies and curtailing model complexity. Nonetheless, with the escalation of polynomial feature degrees, there ensues a commensurate

proliferation in the number of predictors, potentially exacerbating issues of multicollinearity. Such a phenomenon may impinge upon the efficacy of regularization techniques. To delineate the nuanced trade-offs inherent in varying degrees of polynomial features and ascertain the optimal degree conducive to minimizing MSPE, the model underwent an iterative exploration spanning degrees one (1) to three (3). This systematic inquiry into polynomial feature degrees was conducted under the framework of ten-fold cross-validation, ensuring robustness and reliability in the evaluation process. Subsequent analysis revealed that a degree of three (3) yielded the most favorable MSPE outcomes for both lasso and ridge regressions executed with polynomial features.

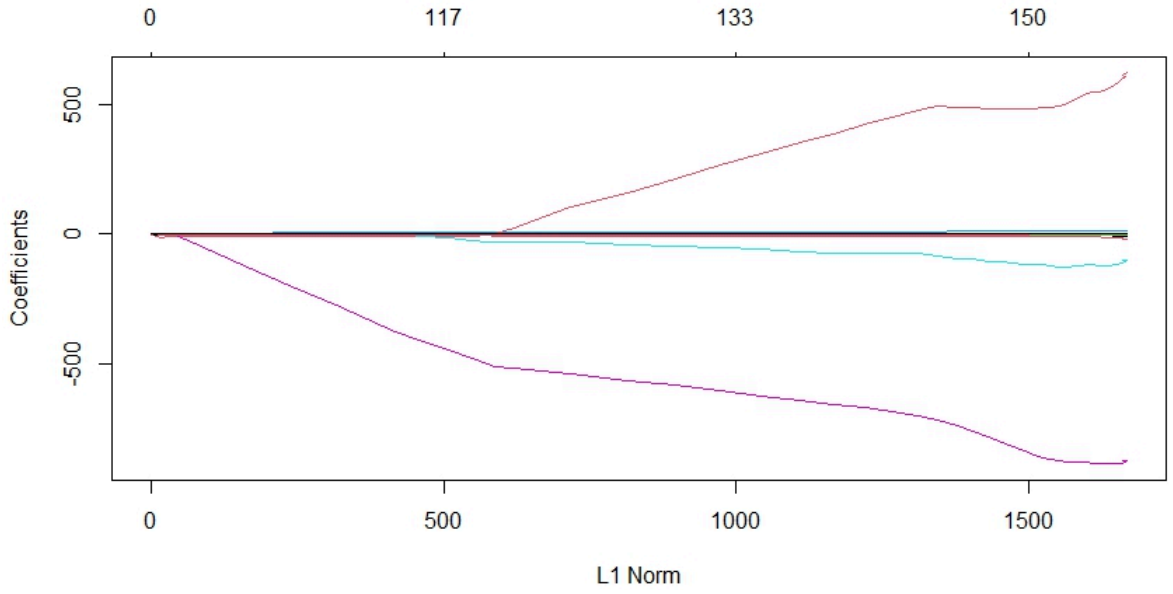

Figure 7: Cross-Validated Lasso Regression Fit

The standardized lasso and ridge regression models, when applied to linear features, exhibit comparable performance across training and testing phases, as evidenced by their close MSPE values under ten-fold cross-validation. However, lasso achieves a marginally lower testing MSPE compared to ridge, underscoring its effectiveness in feature selection which isolates the most predictive variables and enhances generalization.

Upon the introduction of polynomial features, the performance divergence between lasso and ridge does not exhibit a pronounced shift. For ridge regression, the introduction of polynomial features of degree three (3) results in a notable improvement in testing MSPE relative to the linear model. This enhancement reflects ridge's ability to effectively manage the complexity introduced by polynomial terms, leveraging its L2 regularization to mitigate potential multicollinearity and overfitting. Lasso regression maintains a relatively stable performance with the inclusion of polynomial features. The minimal fluctuation in testing

MSPE between linear and polynomial models suggests that lasso's feature selection mechanism adeptly manages the additional complexity associated with higher-order polynomials. By enforcing sparsity in the coefficients, lasso ensures consistent model performance, even as polynomial degrees increase.

3.3 STEPWISE

Stepwise regression is a methodical iterative process that aims to optimize model performance by enhancing both predictive accuracy and interpretability through systematic inclusion or exclusion of predictors. The primary variants of stepwise regression are forward selection, backward elimination, and bi-directional (or both ways) selection.

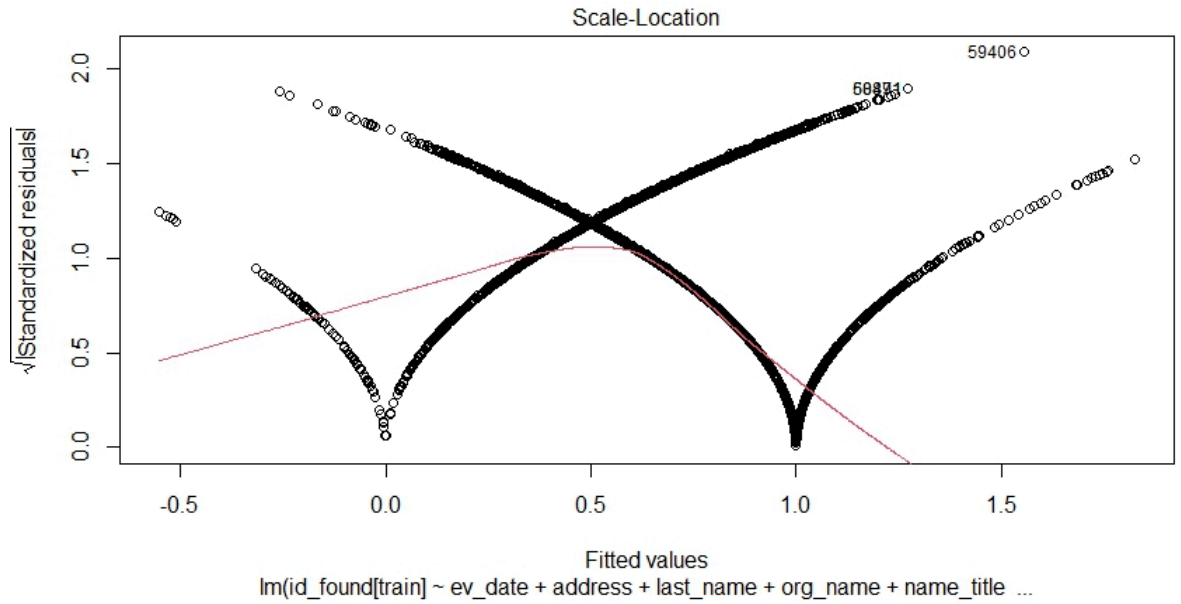

Figure 8: Cross-Validated Bi-Directional Stepwise Scale Location

Forward selection initiates with a model that contains no predictors. Variables are added sequentially based on their contribution to model fit or reduction in statistical error, with the process continuing until no additional variables significantly improve the model or all candidates have been assessed. This approach is beneficial when starting with a large number of potential predictors, as it methodically incorporates variables deemed important. Conversely, backward elimination begins with a model encompassing all potential predictors. Variables are iteratively removed based on criteria such as statistical insignificance or minimal impact on model performance. The procedure proceeds until only significant predictors remain or further removal does not enhance the model. This method is advantageous for simplifying an initially comprehensive model. Bi-directional selection integrates aspects of both forward selection and backward elimination, beginning with the forward inclusion of predictors,

but also allowing for the simultaneous removal of variables that become insignificant as new ones are added. This method facilitates a dynamic adjustment of the model by evaluating both the inclusion and exclusion of predictors throughout the iterative process.

The forward stepwise regression method, characterized by its incremental inclusion of predictors, initially appeared promising. However, the subsequent cross-validation revealed a significant discrepancy, suggesting that while the forward selection process might efficiently identify relevant variables, it may also be prone to overfitting. This overfitting is indicative of a model that, while initially optimized, fails to generalize effectively to new, unseen data. The results highlight a potential shortfall in the model's capacity to maintain robustness across different subsets of the data, thereby impacting its predictive performance. In comparison, the backward stepwise regression method, which operates by sequentially removing predictors, displayed similar patterns of performance. While it initially achieved a favorable evaluation, the cross-validation results suggest that this method too may suffer from overfitting or insufficient complexity reduction. The consistent high MSPE observed during testing underscores a broader issue: the inability of the backward stepwise approach to achieve a balance between model complexity and predictive accuracy. The bi-directional stepwise regression—which amalgamates elements from both forward and backward selection processes—produced high testing MSPE, indicating that the bi-directional approach, similar to its unidirectional counterparts, may be compromised by overfitting. The model's inability to generalize effectively to new data suggests that the bi-directional approach, while comprehensive, does not inherently resolve the challenges associated with variable selection and model stability.

The results from these stepwise methods reveal a common challenge: while these techniques offer systematic approaches to variable selection, they are frequently undermined by issues of overfitting and inadequate model generalization. The high computational demands of stepwise methods further exacerbate their limitations, making them less viable compared to alternative techniques such as lasso and ridge regression. These alternatives provide more efficient and robust approaches to managing variable complexity and achieving model stability, particularly in the context of extensive datasets and complex predictor interactions.

3.4 HIGH DEGREE SPLINE

High-degree spline regression is an advanced modeling technique designed to capture intricate nonlinear relationships between predictors and the target variable. This approach employs splines—piecewise polynomial functions that are seamlessly joined at specific points known as knots. By fitting splines of

elevated polynomial degrees, the model gains enhanced flexibility, enabling it to approximate complex, nonlinear trends within the data.

In evaluating the applicability of high-degree spline regression, it is crucial to account for the inherent characteristics of the predictive variables previously analyzed. The data exhibits two potentially nonlinear interactions between predictors and the target variable. The ability of high-degree spline regressions to flexibly adapt to such complexities, offer a potential solution to accurately represent these relationships.

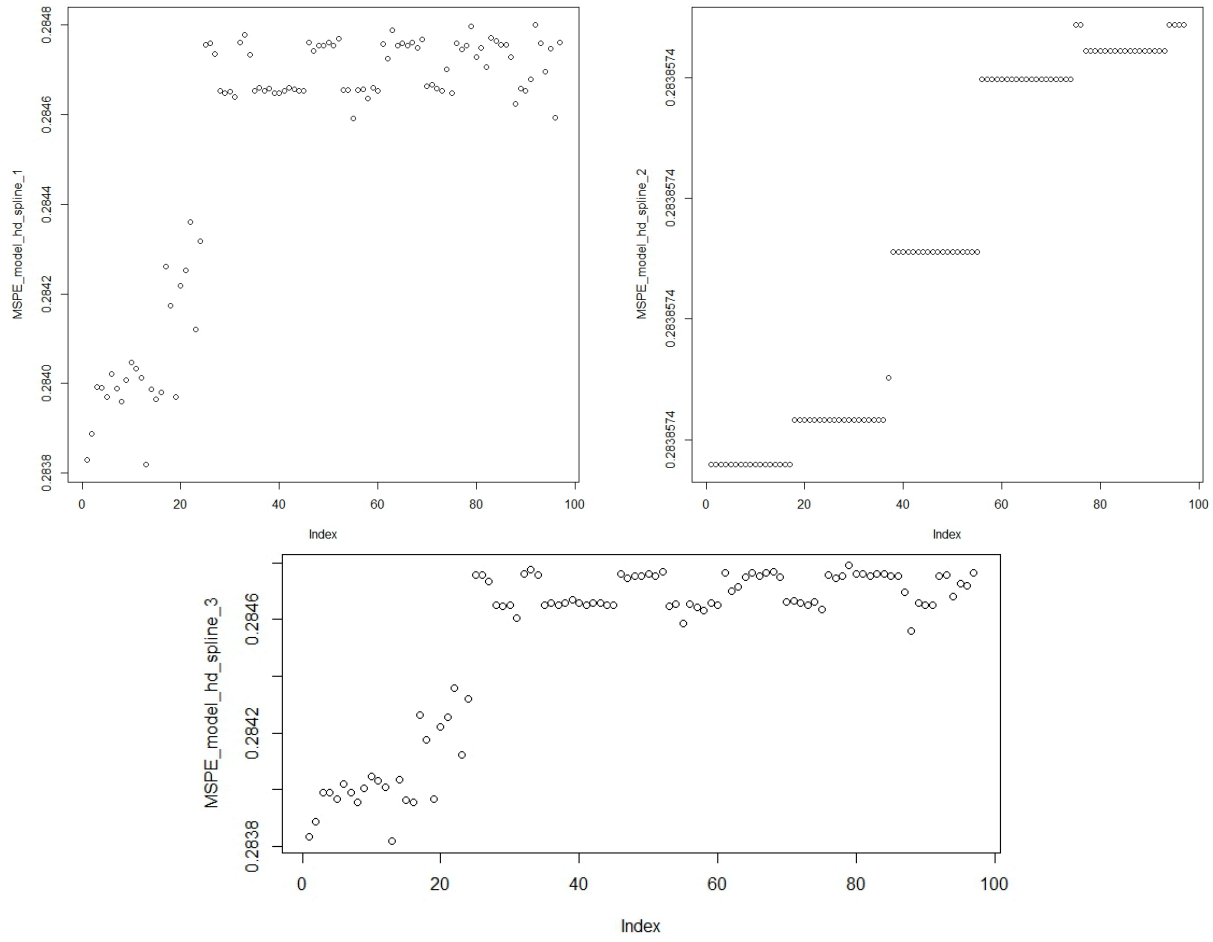

Figure 9: Degrees of Freedoms trade offs High Degree Spline Models.

To ensure that the model achieves an optimal equilibrium between complexity and fit, ten-fold cross-validation was utilized alongside an iterative examination of the model's degrees of freedom—ranging from one (1) to a hundred (100). Figure 9 illustrates the progressive escalation in MSPE as the degrees of freedom surpass twenty, thereby pinpointing the optimal degree of freedom for minimizing MSPE. This approach, by reducing the MSPE, enhances the model's smoothness and its capacity to generalize to unseen data, mitigating the risk of overfitting while preserving a robust representation of the underlying data patterns.

Model 1: High Degree Spline on ev_date, Event Date

This model aimed to uncover the intricate patterns in the relationship between event date and the target variable. The model's testing performance exhibited a stark divergence, with a marked increase in testing MSPE compared to training, signaling a failure to generalize well. Despite the ability to capture nuances in the event date's relationship with the target, the high degrees of freedom of fifty-two (52) led to overfitting, suggesting that the model became too attuned to the idiosyncrasies of the training data.

Model 2: High Degree Spline on address, Number of address fields filled

In applying a high-degree spline to the number of address fields filled, the model similarly attempted to account for the nonlinear associations within the data. However, the performance gap between training and testing was even more pronounced than the high degree spline for *ev_date*. While the model performed adequately on the training data, the sharp rise in testing MSPE exposed considerable overfitting. The disparity between the degrees of freedom for training and testing—seventy-six (76) and two (2) respectively—highlights the difficulty in maintaining a balance between complexity and generalizability. The model's excessive flexibility, indicated by its high degrees of freedom, caused it to overfit the training data, making it unable to perform reliably on unseen data.

Model 3: High Degree Spline on ev_date and address

By incorporating both event date and number of filled address fields, this model introduced additional complexity to account for potential interactions between the two predictors. However, the results pointed to overfitting, as evidenced by the significant difference between training and testing MSPE. The added predictor increased the model's complexity without offering substantial gains in performance, amplifying the challenges in generalizing to new data. The high training degrees of freedom of forty-nine (49) exacerbated these issues, leading to poor testing results and signaling miscalibration of spline parameters.

The high-degree spline regression models provided a mechanism for capturing complex, nonlinear relationships, particularly where traditional linear models fall short. However, their success was limited by the substantial gap between training and testing performance, attributed largely to the models' high degrees of freedom. While splines offer the flexibility needed to model intricate patterns in the data, this flexibility must be carefully constrained to avoid overfitting. In this context, the high degrees of freedom resulted in models that were overly complex, fitting noise in the training data rather than generalizable trends. Compared to other modeling techniques, high-degree splines hold value in cases of pronounced nonlinearity, but require stringent cross-validation and tuning to ensure robust performance across both training and testing datasets.

4—CONCLUSIONS

The final model selection was guided by a thorough evaluation of the Mean Squared Prediction Error (MSPE) across multiple modeling techniques, with particular emphasis on balancing prediction accuracy and model complexity. The lasso regression model with polynomial features of degree three (3) was selected as the optimal model due to its ability to minimize overfitting while maintaining robust predictive performance. This model consistently exhibited the lowest testing MSPE among the candidate models, which was determined by iterating through polynomial degrees and selecting the degree that produced the smallest MSPE.

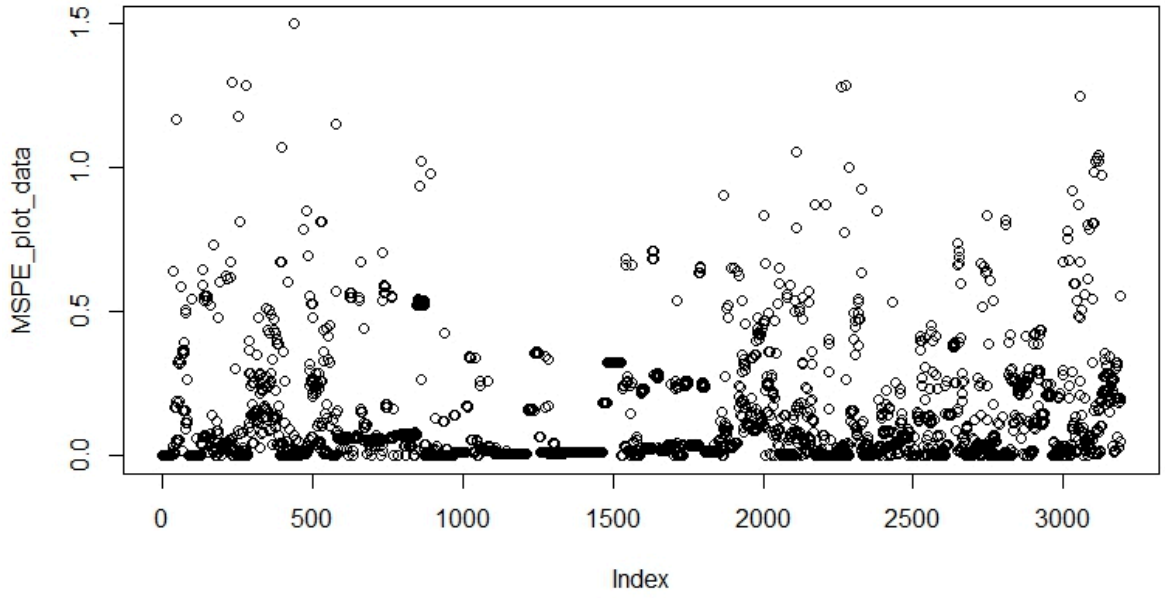

Figure 10: Final Model Error Distribution

The decision to focus on lasso regression with polynomial features was further supported by the distribution of MSPE values across testing samples. Notably, only 22.75% and 18.05% of the test points showed a standard predictive error greater than 0.15 and 0.25 respectively, highlighting the model's relatively strong generalization to unseen data. Compared to other modeling approaches explored—such as high-degree spline regressions and ridge with polynomial features—lasso demonstrated superior feature selection capabilities. Its ability to regularize less meaningful variables while retaining critical predictors ensured a parsimonious model that performed well in reducing prediction errors without adding unnecessary complexity.

The implementation of twenty-fold cross-validation throughout the model selection process provided further confidence in the model's generalizability. By training and testing the model across different data subsets, potential overfitting issues were mitigated, and the most effective polynomial degree was

identified. In this case, degree three (3) was found to capture sufficient nonlinear interactions within the dataset, outperforming both lower- and higher-degree alternatives. Statistical significance was assessed via bootstrapping to ensure the reliability of model coefficients of which is provided in Appendix A.

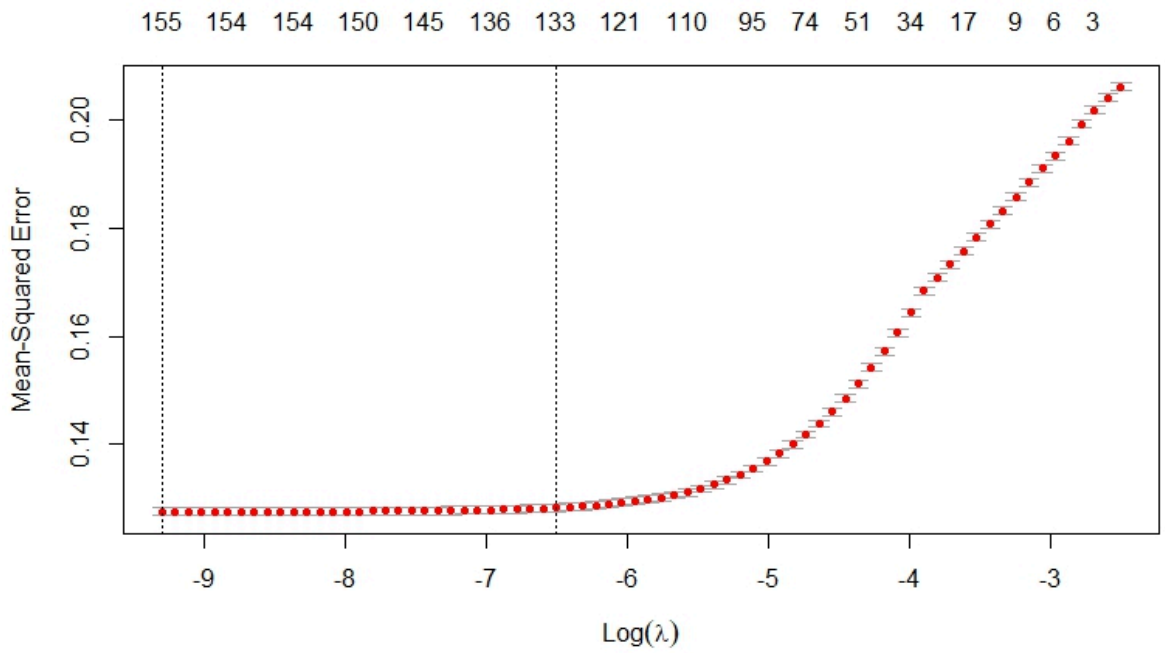

Figure 11: Final Model Cross-Validated Error Curve

The bootstrap procedure, conducted over ten-thousand (10,000) iterations, revealed that the selected model's coefficients were largely significant, reinforcing the model's validity and providing insight into the relative importance of each predictor in the matching process. This approach to model validation not only confirmed the robustness of the selected model, but also underscored the importance of using regularization techniques like lasso when dealing with datasets characterized by high-dimensional features and potential multicollinearity.

4.1 FINAL MODEL RESULTS

The coefficient associated with first names was found to be negative, although it did not achieve statistical significance. This negative coefficient could be attributed to the inherent commonality and variability of first names, which may not provide the requisite specificity for establishing a discernible matching pattern. Nevertheless, due to the lack of statistical significance, no definitive conclusions can be drawn regarding the impact of first names on the matching process. In contrast, the coefficient for last names was positive and statistically significant. This result underscores the critical role of last names in the

matching process, likely attributable to their relatively stable nature and greater uniqueness when in conjunction with first names across records. The statistical significance of this coefficient reinforces the importance of last names as a pivotal element in achieving accurate matches.

The positive and statistically significant coefficient of the *guest_of* variable suggests that being identified as a guest of another constituent enhances the likelihood of a successful match. This finding may initially seem counterintuitive, as guests—frequently parents, guardians, or individuals with no direct affiliation to the university—might be expected to have incomplete or less robust records within the system. However, it is plausible that guests are often linked to sufficient data through their association with the primary invitee or via detailed registration processes. Controlling for registration type also may have further clarified this relationship by mitigating the potential confounding effects of constituencies that share identical contact information. Groups such as minors, who often register with the same details as a parent or guardian, could introduce biases that affect the overall match probability. Controlling for registration type may explain the positive relationship between matching and *guest_of* by more accurately distinguishing between individual registrants and those whose contact information is shared as part of a family or group, reducing the distortion such cases might introduce into the matching process.

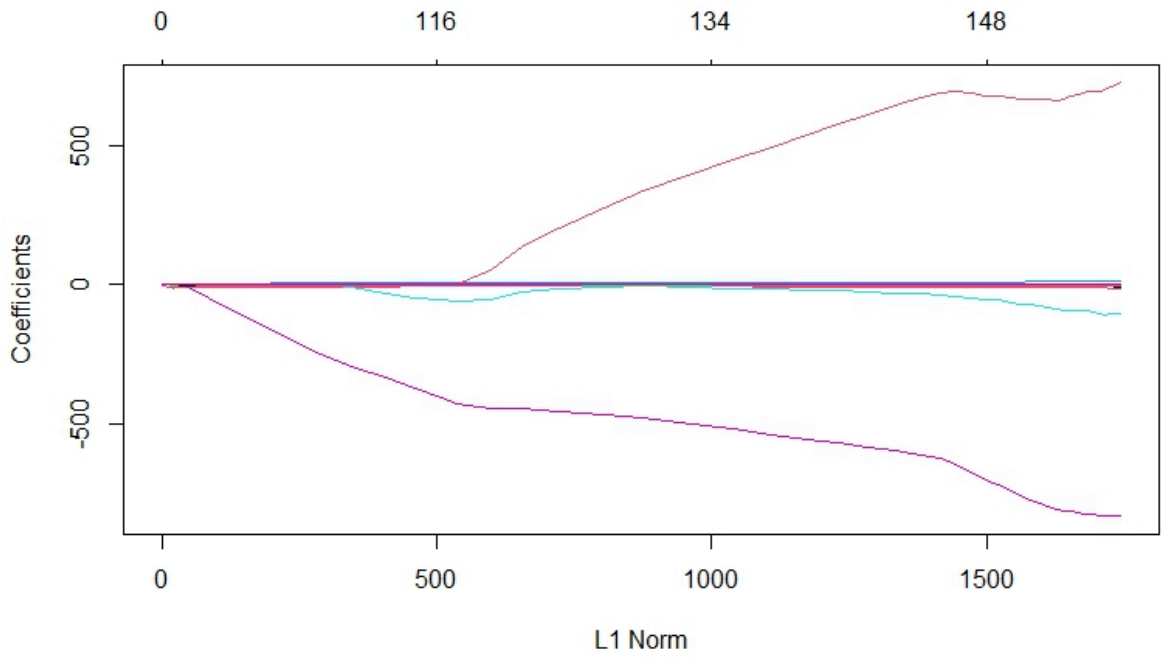

Figure 12: Final Model Fit Performance

Titles or job titles substantially increased the probability of a match, with both variables demonstrating significant positive coefficients. These findings suggest that individuals who provide detailed titles are more likely to have corresponding detailed records in the system, facilitating a successful match. This

might be attributed to the thoroughness with which these constituents fill out registration forms, offering more identifiable information. Similarly, one of the strongest predictors in the model was *email*—indicative of the presence of an email address—which is both large and statistically significant. Given that emails are unique identifiers and commonly required across many platforms, it's no surprise that they outperform other contact information, particularly phone numbers, which were found to be statistically insignificant. However, the lack of significance for phone numbers could be explained by the fact that phone number matching was not actively utilized until March of 2024, along with the fact that many registrants include email addresses when providing a phone number.

Participation metrics such as *no_partic* showed a slight negative impact on matching probability. While the effect was marginal, it could reflect that non-participation in events might signal less engagement with the university, thereby fewer records in the system. However, the change in match performance is small, suggesting that historical engagement does not play as large a role in matching probability.

Academic divisions and graduation years were both strong positive predictors of match probability. The variable *academic_divis*—which denotes a provided affiliation with a UCSD academic division—demonstrated a high positive coefficient, underscoring the significance of this affiliation in enhancing match probability. Since *academic_divis* is exclusively associated with UCSD, its presence is a robust indicator of an individual's history with the university, thereby contributing to higher match success. However, it is important to note that *academic_divis* is inherently reliant on *college_school*, as the latter encompasses affiliations with both UCSD and other institutions. Thus, an academic division will only be provided if *college_school* is also present and marked as UCSD. This indicates that the true effect of providing an academic division is likely less given the casual relationship between the two variables. A possible conclusion to be drawn is that academic division is a strong predictor of matching probability and those with no UCSD affiliation are more likely to negatively impact matching probability.

Constituents with graduation years listed are also far more likely to be matched. Alumni likely have existing university records that include their graduation year. This conclusion is reinforced by the fact that of the registration types, only parent alumni, VIPs, and minors were statistically significant. The positive effect of parent alumni and VIPs aligns with the idea that parents who are also alumni are more likely to have pre-existing records, while VIPs, by nature, are expected to provide accurate and detailed information. COnversely, minors decrease match probability, likely because their records are tied to their parents and not as distinct entities, leading to them being marked as guests or unnamed. The aforementioned analysis of these distinct registration groups suggest that those with university affiliation increase matching probability despite statistically insignificant evidence for the other registration types.

Attendance types displayed some surprising trends. For instance, in person attendance decreased match probability, which is unexpected considering that in-person attendance typically requires providing accurate contact information, such as for ticket distribution. However, when controlling for other variables, the negative coefficient could indicate that the more formal nature of virtual or other remote attendance prompts more precise and consistent data entry, while in-person events might allow for less detailed submissions. On the other hand, on-arrival attendance increased match probability, but the lack of a clear definition of this variable complicates its interpretation. The positive result could stem from more accurate or timely information being provided when someone registers upon arrival, but further investigation into the meaning of this variable would be necessary. Generally, the attendance type variable struggles intuitively and with sample size to provide any useful information as to how attendance mediums affect matching ability.

4.2 EVENT ANALYSIS

Of the one-hundred-thirty-six (136) events analyzed in this study, eighty-seven (87) demonstrated statistically significant effects on match probability. These events were subsequently grouped into categories such as Homecoming, Triton Welcome Week, Alumni 101, and Celebration Weekend. The analyses of event-specific coefficients uncovers distinct patterns within each category, offering insights into the nature of constituent engagement and its impact on match outcomes.

Triton Welcome Week events predominantly exhibited a negative effect on match probability. This outcome is largely attributable to the substantial influx of new students attending these early engagements, many of whom are not yet integrated into ESP. This trend aligns with expectations, as these students, at the outset of their university journey, typically lack established records within the database. Nevertheless, the analysis reveals some variance within this category, suggesting that geographically specific events may outperform their counterparts in generating matches. For instance, Triton Welcome West Coast 2022 significantly exceeded the match rate, indicating that regional targeting may offer a pathway for mitigating the otherwise low match probability of early engagement events. However, this solution is limited by the limited presence of students with existing records during welcome week and the decreased presence of non-freshman prior to week one (1) of Fall quarter.

The trajectory of Alumni Career Network Events presented a more dynamic pattern, initially experiencing a sharp decline in match probability during early 2023, followed by a marked improvement in the latter half of the year. By the close of 2023, these events exhibited a positive correlation with match outcomes, with several events in 2024 yielding particularly high coefficients. This progression may reflect a growing alignment between alumni engagement efforts and the accuracy of ESP records, perhaps attributable to

more refined post-event data capture processes or targeted outreach strategies aimed at alumni with higher match potential. Such trends suggest that these events are increasingly successful in not only fostering alumni participation but also ensuring the integrity and completeness of alumni data within the system.

Event Category	# of Statistically Significant Events	Mean	Median	Q1	Q3	Variance
Triton Welcome	11	0.0361	-0.1542	-0.2381	0.3756	0.1493
Alumni Career Network	9	-0.0672	0.1463	-0.5454	0.5269	0.5992
Alumni 101	8	0.4830	0.4205	0.3404	0.6588	0.0382
Celebration Weekend	2	-0.3241	-0.3241	-0.3688	-0.2794	0.0160
Leaders Conference	6	0.1534	0.0411	-0.1683	0.2612	0.2151
Regional Series	14	0.0781	0.1431	-0.1491	0.2971	0.0670
Homecoming	2	0.0832	0.0832	-0.0669	0.2332	0.1800
Triton Table Talk	6	-0.2228	-0.2589	-0.2909	-0.0835	0.1117
International	7	-0.6337	-0.7984	-0.8705	-0.5391	0.2107
Take a Triton	7	0.2173	0.2734	0.2005	0.3096	0.0570
Misc	15	-0.0343	0.1190	-0.2648	0.2506	0.1683

Table 3: Event Coefficients by Category Statistics

Consistently outperforming other event categories, Alumni 101 events demonstrated a robust positive influence on match probability across all analyzed periods. These events, which focus on facilitating interaction between current students and alumni professionals, appear to attract alumni with deeper institutional ties—an essential factor for accurate matching. The consistently strong coefficients support the hypothesis that alumni engaged in career mentorship or professional networking are more likely to maintain current contact information, thus driving higher match rates. This positive relationship between career-focused engagement further reinforces the value of such events in both alumni relations and data management.

In contrast, Celebration Weekend events were associated with a negative impact on match probability, as reflected by consistently negative coefficients in both 2022 and 2023. These findings suggest that social or celebratory events may draw a broader and more heterogeneous audience, many of whom may not maintain up-to-date records or have less consistent interaction with the alumni system. Given the relatively limited longitudinal data available for these events, it remains plausible that future iterations

may reveal more nuanced insights into their matchability, though the current trend indicates a clear challenge in maintaining accurate records for these constituents. A similar downward trend was observed in Triton Table Talk events, which consistently demonstrated a reduction in match probability of at least 20% across all analyzed periods. While recent data from 2024 suggests a slight improvement, the informal and conversational nature of these events may inherently limit the provision of accurate or updated contact information, leading to persistently lower match rates.

The domestic Regional Series events—comprising Fall, Winter, and Spring events—generally contributed to increased match rates, though seasonal fluctuations were noted. Specifically, match rates tended to decline between March and July, a trend that merits further investigation. Nevertheless, most events within this category demonstrated positive effects on matchability. The observed seasonal variations may indicate that external factors—such as the academic calendar or event timing—play a role in shaping the success of these events in achieving accurate matching.

In stark contrast, international Regional Series events have shown a consistent decline in match probability since 2022. This pattern suggests that international constituents face greater barriers, likely due to lower levels of institutional engagement or discrepancies in data capture processes for international attendees. The uniform decline in positive impacts on match probability across international events highlights the need for targeted interventions to improve the matchability of international alumni, whose records appear to be more vulnerable to inaccuracies or gaps in data collection.

Conversely, Take a Triton to Work events consistently produced positive effects on match probability. The nature of these events, which rely on accurate contact information to facilitate professional connections between students and alumni, likely explains the high match rates observed. The structured, purpose-driven format of these events appears to incentivize both alumni and participants to provide and maintain up-to-date constituent data, further underscoring the importance of such engagements in fostering accurate record-keeping within ESP.

On the other hand, the analysis of Homecoming events proved inconclusive due to the limited availability of statistically significant second-level event data. The primary Homecoming event in 2022 yielded a negative coefficient (-0.217), yet follow-up registration data for the same year suggested a more favorable outcome, with a positive coefficient of 0.383. This dichotomy indicates that follow-up events may benefit from improved data collection efforts and pre-existing constituent records, though further investigation is needed to confirm this trend. Similarly, the analysis of Leadership Conference events revealed no consistent trend, with match probabilities oscillating between positive and negative effects. Nonetheless, a notable outlier, the Triton Leaders Conference Social in early 2024, yielded an exceptionally high

coefficient of 0.992. This significant deviation warrants further exploration to uncover the underlying factors contributing to such a high improvement of matching success, as well as its broader implications for leadership-focused events.

4.3 FINAL CONCLUSIONS

Based on the analysis of the final model, it is clear that event type and active university engagement play a critical role in influencing the probability of matching an alumni event participant to an existing ESP constituent. The findings reveal that the nature of the event—whether it is career-oriented, celebratory, or informal—significantly impacts the likelihood of accurate matching, with structured and purposeful engagements yielding more favorable outcomes. Events that necessitate precise contact information, such as career development activities and alumni mentoring programs, demonstrate higher match probabilities. This correlation suggests that when events require or incentivize accurate data submission, there is a corresponding improvement in match rates.

Conversely, the presence or characteristics of contact information and attendees themselves appear to have a limited effect on match probability. This highlights a potential issue: the effectiveness of event-driven data collection is often contingent upon the inherent incentives for accuracy. In the absence of mechanisms such as pre-populated forms or closed-ended registration questions that encourage precise submissions, there is diminished motivation for attendees to provide unique information, particularly if the event does not actively engage them or necessitate this information. Effective matching hinges not only on the type of event, but also on the strategic alignment of engagement practices with data accuracy requirements.

5—WORKS REFERENCED

Sari, Bruno Giacomini, et al. “Interference of Sample Size on Multicollinearity Diagnosis in Path Analysis.” Pesquisa Agropecuária Brasileira, Embrapa Secretaria de Pesquisa e Desenvolvimento; Pesquisa Agropecuária Brasileira, 1 June 2018, doi.org/10.1590/S0100-204X2018000600014.

Žiga Pušnik, Miha Mraz, Nikolaj Zimic, Miha Moškon, Review and assessment of Boolean approaches for inference of gene regulatory networks, Heliyon, Volume 8, Issue 8, 2022, e10222, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2022.e10222.

6—APPENDIX

APPENDIX A—FINAL MODEL OUTPUT

The table below contains the coefficients for the final model which is a third-degree polynomial lasso regression with twenty-cross-validations. The lower limit and upper limit for the confidence interval calculated using bootstrapping is provided along with the boolean indicating statistical significance based on that confidence interval.

Variable Name	Coefficient	Lower_CI	Upper_CI	Statistical Significance
1.0	-9.400	-15.730	-0.001	TRUE
2.0	-26.677	-34.123	-13.315	TRUE
3.0	0.579	-5.107	11.787	FALSE
0.1	11.180	9.769	13.550	TRUE
1.1	-18.052	-1587.173	192.969	FALSE
2.1	-807.083	-1117.424	525.255	FALSE
0.2	1.579	0.224	3.540	TRUE
1.2	732.679	245.753	1150.049	TRUE
0.3	-1.167	-2.382	0.149	FALSE
first_name	-0.024	-0.248	0.179	FALSE
last_name	0.249	0.052	0.471	TRUE
org_name	0.030	0.007	0.056	TRUE
name_title	0.210	0.170	0.251	TRUE
job_title	0.200	0.178	0.216	TRUE
email	0.345	0.331	0.366	TRUE
phone	0.000	-0.014	0.009	FALSE
is_invitee	0.001	-0.012	0.022	FALSE
is_guest	0.004	-0.009	0.023	FALSE
guest_of	0.035	0.022	0.047	TRUE
invite_accepted	0.003	-0.006	0.010	FALSE
invite_rejected	0.022	-0.045	0.074	FALSE
yes_partic	-0.008	-0.023	0.003	FALSE
no_partic	-0.016	-0.033	-0.005	TRUE
cancel_date	0.085	0.012	0.131	TRUE
academ_divis	0.910	0.896	0.928	TRUE
college_school	-0.688	-0.705	-0.672	TRUE
grad_yr	0.310	0.300	0.319	TRUE
reg_date	-0.114	-0.126	-0.101	TRUE

event_id_ACE TWE Asia 20220813	0.520	0.411	0.632	TRUE
event_id_ACE TWE East Coast 20220808	0.479	0.370	0.590	TRUE
event_id_ACE TWE West Coast 20220815	0.720	0.618	0.817	TRUE
event_id_ACE_ All UC Alumni Career Network: Networking Like a Pro_20240313	0.527	0.423	0.579	TRUE
event_id_ACE_45th Annual Alumni Awards East Coast_20240328	0.522	0.413	0.575	TRUE
event_id_ACE_45th Annual Alumni Awards West Coast_20240321	0.355	0.245	0.423	TRUE
event_id_ACE_All UC Alumni Career Network | Retiring the Concept of Retirement_20230621	-0.054	-0.195	0.050	FALSE
event_id_ACE_All UC Alumni Career Network Webinar: Being a Seasoned Professional in a New Workplace_20221114	0.091	-0.002	0.179	FALSE
event_id_ACE_All UC Alumni Career Network Webinar: Enhancing Your Career Journey Through Mentorship_20230126	0.146	0.053	0.228	TRUE
event_id_ACE_All UC Alumni Career Network: ...Careers in Environmental Justice_20240207	0.418	0.306	0.481	TRUE
event_id_ACE_All UC Alumni Career Network: AI's Impact on Jobs: Uncertainty and Opportunities_20240131	0.668	0.573	0.722	TRUE
event_id_ACE_All UC Alumni Career Network: How to Land a Job at the UC_20230509	-0.241	-0.378	-0.071	TRUE
event_id_ACE_Alumni 101: Advancing Your Career Journey Through Mentorship_20240123	0.341	0.243	0.394	TRUE
event_id_ACE_Alumni 101: Career Search Success: Harnessing AI Tools with a Human Touch_20240404	0.642	0.523	0.704	TRUE
event_id_ACE_Alumni 101: From Campus to Career: Building a Strong Alumni Network_20240522	0.373	0.242	0.433	TRUE
event_id_ACE_Alumni 101: How to Use Your Alumni Network for Career Success_20231025	0.709	0.625	0.781	TRUE
event_id_ACE_Alumni 101: Leveraging Informational Interviews as a Job Search Strategy_20231010	0.234	0.147	0.305	TRUE
event_id_ACE_Alumni 101: Tips for Career Fair Success_20231003	0.339	0.232	0.430	TRUE

event_id_ACE_Alumni 101: What is Experiential Learning and Why is it Important_20240213	0.758	0.614	0.868	TRUE
event_id_ACE_Alumni 101: Why Your First Job Matters_20240227	0.468	0.368	0.531	TRUE
event_id_ACE_Alumni Career Networking Reception_20231011	0.772	0.642	0.920	TRUE
event_id_ACE_Alumni Celebration Weekend 2023_FY23	-0.414	-0.502	-0.332	TRUE
event_id_ACE_Alumni Celebration Weekend_FY22	-0.235	-0.344	-0.127	TRUE
event_id_ACE_Alumni Opportunity at Commencement_20240615	0.284	0.129	0.388	TRUE
event_id_ACE_APIAC: Triton Leaders Conference Social_20240203	0.992	0.859	1.062	TRUE
event_id_ACE_Black Achievement and Contributions Kick Off Alumni_20240305	0.423	0.302	0.500	TRUE
event_id_ACE_Day 1 Triton Leaders Conference 2023	0.280	0.185	0.377	TRUE
event_id_ACE_Day 2 Triton Leaders Conference 2023	0.206	0.111	0.289	TRUE
event_id_ACE_Fall Regional Celebrations: New York_20221027	0.518	0.411	0.615	TRUE
event_id_ACE_Fall Regional Celebrations: San Francisco_20221207	0.345	0.246	0.428	TRUE
event_id_ACE_Fall Regional Celebrations: Washington D.C._20221025	-0.011	-0.127	0.100	FALSE
event_id_ACE_Fall Regional Series: Bay Area_20231005	-0.038	-0.136	0.030	FALSE
event_id_ACE_Fall Regional Series: DC_20230928	0.020	-0.078	0.121	FALSE
event_id_ACE_Fall Regional Series: Los Angeles 20230921	0.047	-0.057	0.121	FALSE
event_id_ACE_Founders Weekend 11/13-11/15, 2014	0.217	0.108	0.336	TRUE
event_id_ACE_GOLD Reunion & Happy Hour_20231020	0.213	0.131	0.276	TRUE
event_id_ACE_Golden Triton Induction Class of 74_20231021	0.119	0.009	0.266	TRUE
event_id_ACE_Homecoming 2022_FY23	0.383	0.284	0.475	TRUE
event_id_ACE_Homecoming_FY22	-0.217	-0.350	-0.074	TRUE
event_id_ACE_Homecoming_FY23	-0.070	-0.171	0.013	FALSE

event_id_ACE_International Regional Series: Delhi_20230113	0.079	-0.031	0.188	FALSE
event_id_ACE_International Regional Series: Mumbai_20230108	0.290	0.193	0.377	TRUE
event_id_ACE_JPM Networking Social_20220109	-0.256	-0.422	-0.116	TRUE
event_id_ACE_LionTree Celebration 20221112	0.134	0.023	0.256	TRUE
event_id_ACE_New Triton Welcome San Francisco_20230722	0.013	-0.075	0.084	FALSE
event_id_ACE_New Triton Welcome: Asia_20240717	-0.168	-0.333	-0.111	TRUE
event_id_ACE_New Triton Welcome: Bay Area_20240727	-0.022	-0.156	0.015	FALSE
event_id_ACE_New Triton Welcome: India_20240717	0.273	0.107	0.334	TRUE
event_id_ACE_New Triton Welcome: Los Angeles_20240714	-0.002	-0.127	0.046	FALSE
event_id_ACE_New Triton Welcome: San Diego_20240713	-0.392	-0.520	-0.354	TRUE
event_id_ACE_New Triton Welcome: United States & Europe_20240724	-0.154	-0.309	-0.111	TRUE
event_id_ACE_Regional Celebrations: Los Angeles_20230321	-0.096	-0.193	-0.019	TRUE
event_id_ACE_San Deigo Club: Yoga and Happy Hour_20230312	-0.938	-1.083	-0.786	TRUE
event_id_ACE_Spring Regional Series San Diego_20240602	-0.313	-0.431	-0.273	TRUE
event_id_ACE_Spring Regional Series: Los Angeles_20220524	-0.149	-0.278	-0.017	TRUE
event_id_ACE_Spring Regional Series: Orange County_20220525	-0.082	-0.254	0.043	FALSE
event_id_ACE_Spring Regional Series: Sacramento_20240411	-0.220	-0.339	-0.143	TRUE
event_id_ACE_Spring Regional Series: San Diego_20220426	-0.184	-0.313	-0.040	TRUE
event_id_ACE_Spring Regional Series: San Jose_20240409	0.326	0.196	0.392	TRUE
event_id_ACE_Spring Regional Series: Seattle_20240514	0.160	0.030	0.239	TRUE
event_id_ACE_Spring Regionals San Diego_20230413	0.191	0.093	0.282	TRUE
event_id_ACE_Spring Regionals Seattle_20230418	-0.148	-0.257	-0.049	TRUE

event_id_ACE_Spring Regionals: Boston_20230604	0.327	0.230	0.405	TRUE
event_id_ACE_Spring Regionals: New York_20230601	0.211	0.122	0.294	TRUE
event_id_ACE_Spring Regionals: Sacramento_20230412	0.126	0.001	0.227	TRUE
event_id_ACE_Spring Regionals: San Diego_20230413	-0.052	-0.151	0.030	FALSE
event_id_ACE_Spring Regionals: Washington DC_20230606	-0.123	-0.269	0.006	FALSE
event_id_ACE_Take a Triton to Class 2022	0.308	0.210	0.393	TRUE
event_id_ACE_Take a Triton to Class Fall 2022_FY23	0.256	0.151	0.343	TRUE
event_id_ACE_Take a Triton to Class: Engaging with a Changing Planet_20240129	0.311	0.197	0.392	TRUE
event_id_ACE_Take a Triton to Class: How Environmental Law Affects Your Life in San Diego_20240131	0.093	-0.017	0.162	FALSE
event_id_ACE_Take a Triton to Class: How to Practice Emotional Resilience_20231016	0.075	-0.030	0.145	FALSE
event_id_ACE_Take a Triton to Class: Nutrition for a Healthy Lifestyle_20231016	-0.270	-0.392	-0.191	TRUE
event_id_ACE_Take a Triton to Class: Understanding the Partisan Brain_20231018	-0.067	-0.171	0.010	FALSE
event_id_ACE_Take a Triton to Class: Using Community-Based Science to Study the Ecology of Marshes_20240129	0.273	0.165	0.344	TRUE
event_id_ACE_Take a Triton to Work Match Day_20231107	0.497	0.418	0.554	TRUE
event_id_ACE_Take a Triton to Work Match Day: WI/SP_20240220	0.145	0.061	0.186	TRUE
event_id_ACE_Triton 5K_20221023	-0.290	-0.390	-0.205	TRUE
event_id_ACE_Triton 5K_20231022	0.070	-0.009	0.123	FALSE
event_id_ACE_Triton Day, Sn Diego_20150404	-0.273	-0.421	-0.161	TRUE
event_id_ACE_Triton Fan Celebration - Irvine_20240118	-0.095	-0.264	0.017	FALSE
event_id_ACE_Triton Fan Celebration MVB v UCLA_20240126	0.003	-0.141	0.111	FALSE
event_id_ACE_Triton Leaders Conference Day 1 (Virtual)_20220204	-0.183	-0.301	-0.053	TRUE
event_id_ACE_Triton Leaders Conference Day 2 (In Person)_20220205	-0.251	-0.374	-0.126	TRUE

event_id_ACE_Triton Leaders Conference_2024	-0.124	-0.215	-0.084	TRUE
event_id_ACE_Triton Table Talk | Building and Measuring Cultural Impact_20240423	-0.232	-0.377	-0.178	TRUE
event_id_ACE_Triton Table Talk | Women In Wealth: Giving for Good_20240319	-0.034	-0.149	-0.005	TRUE
event_id_ACE_Triton Table Talk: Driving Change: Careers in Sustainability_20240220	0.260	0.149	0.333	TRUE
event_id_ACE_Triton Table Talk: Investing with Confidence: ...Long Term Success_20240512	0.004	-0.104	0.050	FALSE
event_id_ACE_Triton Table Talk: Keys to Thriving at UC San Diego and Beyond_20231021	-0.753	-0.835	-0.672	TRUE
event_id_ACE_Triton Table Talk: Perspectives from Financial Industry Leaders_20230919	-0.017	-0.105	0.043	FALSE
event_id_ACE_Triton Table Talk: Perspectives from Financial Industry Leaders_20310919	-0.293	-0.430	-0.136	TRUE
event_id_ACE_Triton Tailgate_20221021	-0.212	-0.311	-0.127	TRUE
event_id_ACE_Trtton Table Talk: Supporting Success: Women in Wealth Explored_20230305	-0.286	-0.391	-0.178	TRUE
event_id_ACE_TWE Europe_20220815	-0.212	-0.326	-0.104	TRUE
event_id_ACE_TWE P&F East Coast_20220806	-0.068	-0.180	0.046	FALSE
event_id_ACE_TWE P&F West Coast_20220820	0.043	-0.061	0.161	FALSE
event_id_ACE_TWE San Diego_20220814	-0.278	-0.381	-0.177	TRUE
event_id_ACE_UC ALUMNI CAREER NETWORK | Perspectives on Quiet Quitting_20231018	0.001	-0.114	0.087	FALSE
event_id_ACE_UC ALUMNI CAREER NETWORK | Redefining Success: The First-Generation College Experience_20231108	-0.097	-0.271	0.032	FALSE
event_id_ACE_UC Alumni Career Network: HireUC Alumni Career Fair_20230824	-0.859	-0.977	-0.776	TRUE
event_id_ACE_UC Alumni Career Network: Professional Etiquette Post Pandemic_20230809	-0.545	-0.662	-0.440	TRUE
event_id_ACE_UC Alumni Career Network: Social Mobility and Adjusting to an Office Culture_20230718	-1.491	-1.678	-1.336	TRUE
event_id_ACE_UC San Diego International: Dubai_20231026	-0.419	-0.617	-0.257	TRUE

event_id_ACE_UC San Diego International: Hong Kong_20231106	-0.858	-1.053	-0.723	TRUE
event_id_ACE_UC San Diego International: Mumbai_20231029	-1.108	-1.244	-0.956	TRUE
event_id_ACE_UC San Diego International: Mumbai_20240425	-0.883	-0.992	-0.846	TRUE
event_id_ACE_UC San Diego International: New Delhi_20240424	-0.659	-0.789	-0.572	TRUE
event_id_ACE_UC San Diego International: Singapore_20231108	-0.798	-1.033	-0.621	TRUE
event_id_ACE_UCSD Alumni & Friends Mixer at J.P. Morgan Healthcare Conference_20240107	-0.167	-0.298	-0.105	TRUE
event_id_ACE_Unidos X Siempre: CLAC Mentor Program Mixer_20231013	0.025	-0.086	0.106	FALSE
event_id_New Triton Welcome Los Angeles	-0.264	-0.351	-0.186	TRUE
event_id_New Triton Welcome San Diego	-0.126	-0.216	-0.054	TRUE
event_id_PAR_P&F Breakfast & Student Research Showcase_20221022	-0.645	-0.747	-0.551	TRUE
event_id_Silicon Valley GOLD: Pre-Event Meetup	-0.002	-0.207	0.089	FALSE
reg_Academic	0.045	0.005	0.109	TRUE
reg_Alumni	0.000	-0.014	0.029	FALSE
reg_Community Member	-0.007	-0.024	0.024	FALSE
reg_Minor	-0.082	-0.105	-0.048	TRUE
reg_NA	0.000	-0.016	0.033	FALSE
reg_Parent/Alumni	0.124	0.041	0.253	TRUE
reg_Parent/Family	0.012	0.000	0.044	FALSE
reg_Staff/Faculty	0.002	-0.015	0.035	FALSE
reg_Staff/Faculty Alumni	0.001	-0.020	0.036	FALSE
reg_Student	-0.017	-0.032	0.011	FALSE
reg_STUDENT ORG	0.016	-0.039	0.071	FALSE
reg_VIP	0.117	0.026	0.202	TRUE
atten_type_In Person	-0.196	-0.231	-0.173	TRUE
atten_type_Live	0.000	-0.035	0.037	FALSE
atten_type_NA	0.021	0.000	0.039	FALSE
atten_type_On-Demand	-0.153	-0.305	-0.010	TRUE
atten_type_OnArrival	0.074	0.048	0.096	TRUE

Table 4: Final Model Coefficients & Statistical Significance.

APPENDIX B—REGISTRATION TYPE STATISTICS

This table outlines the quantities of each registration type used as a dummy variable. Those registration types that did not pass the threshold filter as outlined in the data processing section were removed during.

Registration Type	Total	# of Matched of Type	% of Type
Community Member	4,154	2,751	66.23%
Parent/Family	15,494	10,969	70.80%
Alumni	12,535	8,812	70.30%
Student	17,881	13,070	73.09%
VIP	85	66	77.65%
Staff/Faculty	2,726	1,972	72.34%
Staff/Faculty Alumni	1,317	969	73.58%
Parent/Alumni	37	34	91.89%
Current Student	15	11	73.33%
Parent Staff/Faculty Alumni	7	6	85.71%
Alumni/Staff	13	9	69.23%
Alumni/Student	11	7	63.64%
STUDENT ORG	157	121	77.07%
Other	5	3	60.00%
Parent Staff/Faculty	17	13	76.47%
Friend	1	1	100.00%
Academic	135	121	89.63%
Minor	1,355	815	60.15%
Staff	9	7	77.78%
Alumni/Faculty	1	1	100.00%
New Grad	2	0	0.00%

Table 5: Registration Type Quantities from Raw Data

APPENDIX C—BINARY VARIABLE SUMMARY QUANTITIES

The table provides the quantities of each binary variable. The second column is the total number of that variable when the boolean is positive and the adjacent third column is when the boolean is true and the constituent was matched in ESP. Subsequent columns five (5) and six (6) are the inverse.

Variable Name	True	True & Match	Match Density	False	False & Match	Match Density
id_stat	0	0	NA	1,203	4	0.33%
source_id	39,410	38,911	98.73%	26,135	6,615	25.31%
id_extracted	29,084	29,068	99.94%	36,461	16,458	45.14%
first_name	63,608	45,076	70.87%	1,937	450	23.23%
last_name	63,592	45,069	70.87%	1,953	457	23.40%
org_name	3,440	2,305	67.01%	62,105	43,221	69.59%
name_title	453	316	69.76%	65,092	45,210	69.46%
job_title	5,039	3,950	78.39%	60,506	41,576	68.71%
email	59,313	42,676	71.95%	6,232	2,850	45.73%
phone	27,519	18,391	66.83%	38,026	27,135	71.36%
reg	0	0	NA	9,588	5,768	60.16%
is_invitee	38,112	27,353	71.77%	27,433	18,173	66.25%
is_guest	17,118	11,161	65.20%	48,427	34,365	70.96%
guest_of	52,415	36,803	70.21%	13,130	8,723	66.44%
invite_accepted	40,109	28,922	72.11%	25,436	16,604	65.28%
invite_rejected	170	106	62.35%	65,375	45,420	69.48%
yes_partic	19,193	12,969	67.57%	46,352	32,557	70.24%
no_partic	24,841	15,930	64.13%	40,704	29,596	72.71%
cancel_date	168	139	82.74%	65,377	45,387	69.42%
modif_date	26	26	100.00%	65,519	45,500	69.45%
action_date	2	2	100.00%	65,543	45,524	69.46%
academ_divis	18,873	15,349	81.33%	46,672	30,177	64.66%
college_school	21,183	15,518	73.26%	44,362	30,008	67.64%
grad_yr	31,818	25,038	78.69%	33,727	20,488	60.75%
reg_date	39,592	26,155	66.06%	25,953	19,371	74.64%

Table 6: Variable Quantities & Match Percentages

APPENDIX D—TRAINING DATA CODE

The following contains the R do-file utilized to train the models on the training data.

```
## ------------------------------------------------------------
## Library Installation
library(readxl)
library(ggplot2)
library(ggmosaic)
library(dplyr)
library(tidyr)
library(ggplot2)
library(plotly)
library(fastDummies)
library(splines)
library(glmnet)
library(MASS)
library(openxlsx)
library(boot)

## ------------------------------------------------------------
## Setup
dataOrg <- read_excel("Compiled Data.xlsx", "Raw Data")
dataOrg <- dataOrg[dataOrg$id_stat != "HELP", ]
#dataOrg$reg[dataOrg$reg == "NA"] <- NA
#dataOrg$atten_type[dataOrg$atten_type == "NA"] <- NA
dataOrg <- dataOrg[!is.na(dataOrg$event_id), ]
dataOrg <- dataOrg[c("id_found", setdiff(names(dataOrg), "id_found"))]

names(dataOrg)
head(dataOrg)
summary(dataOrg)

continuous_X <- subset(dataOrg, select = c(ev_date, address))
dummy_x <- subset(dataOrg, select = c(event_id, reg, atten_type))
binary_X <- subset(dataOrg, select = c(reviewed, source_id, id_extracted, first_name, last_name,
org_name, name_title, job_title, email, phone, is_invitee, is_guest, guest_of, invite_accepted,
invite_rejected, yes_partic, no_partic, cancel_date, modif_date, action_date, academ_divis,
college_school, grad_yr, reg_date))
id_found <- dataOrg$id_found


## ------------------------------------------------------------
## DATA PROCESSING

## Drop ID_Stat
dataOrg <- subset(dataOrg, select = -id_stat)

## Date Value Converted (now number of days since a 1/1/2000)
```

```
dataOrg$ev_date <- as.numeric(as.Date(dataOrg$ev_date) - as.Date("2000-01-01"))
filtered_data <- subset(dataOrg, reviewed > 0)

# Combining Variables
#filtered_data <- filtered_data %>%
#  mutate(full_name = ifelse(fir_name > 0 & las_name > 0, 1, 0))

data_with_dummies <- dummy_cols(filtered_data)

## Rare Column Removal
boolean_proportions <- colMeans(data_with_dummies == TRUE, na.rm = TRUE)
threshold <- 0.0005
low_variability_columns <- names(boolean_proportions[boolean_proportions < threshold])
low_variability_columns <- setdiff(low_variability_columns, names(continuous_X))
print(low_variability_columns)

## Remove redundant dummy var columns
cols_to_remove <- c('event_id', 'reg', 'atten_type')
cols_to_remove <- c(cols_to_remove, low_variability_columns)

data_pre_multi <- as.data.frame(data_with_dummies[, !names(data_with_dummies) %in%
cols_to_remove])

## Columns to select
dummy_x <- colnames(data_pre_multi)[grepl("^event_id_|^reg_|^atten_type_",
colnames(data_pre_multi))]
dummy_x <- dummy_x[!grepl("^reg_date$", dummy_x)]

continuous_X <- unlist(list('ev_date', 'address'))
binary_X <- unlist(list("reviewed", #"source_id", "id_extracted",
                        "first_name", "last_name", "org_name", "name_title", "job_title", "email", "phone",
"is_invitee", "is_guest", "guest_of", "invite_accepted", "invite_rejected", "yes_partic", "no_partic",
"cancel_date", "modif_date", "action_date", "academ_divis", "college_school", "grad_yr", "reg_date"))

## Remove multi-collinearity
OLS <- lm(id_found ~ ., data = data_pre_multi)
summary(OLS)

keep_var <- names(OLS$coefficients)[!is.na(OLS$coefficients)][-1]
keep <- c('id_found', keep_var)
multi_colin_vars <- gsub("`", "", keep)
#multi_colin_vars <- setdiff(keep, binary_vars_to_remove)
data <- subset(data_pre_multi, select = multi_colin_vars)

## Subset Creation
dummy_x <- intersect(names(data), dummy_x)
continuous_X <- intersect(names(data), continuous_X)
binary_X <- intersect(names(data), binary_X)

dummy_x <- subset(data, select = dummy_x)
```

```
continuous_X <- subset(data, select = continuous_X)
binary_X <- subset(data, select = binary_X)
id_found <- data$id_found
data <- cbind(id_found, binary_X, continuous_X, dummy_x)

removed_var <- setdiff(names(data_with_dummies), multi_colin_vars)
print(removed_var)

#print(any(is.na(data)))
#print(sapply(data, function(x) sum(is.na(x))))
#print(unique(filtered_data$atten_type))


## ------------------------------------------------------------
### Compare models via K-fold Cross Validation
k <- 10
n <- length(id_found)

set.seed(7142)
id <- sample(rep(1:k, length = n))

column_names <- c("Model Name", "MSPE", "Polynomial Degree or Degress of Freedom")
MSPE_df <- data.frame(matrix(ncol = length(column_names), nrow = 0))
colnames(MSPE_df) <- column_names

## ------------------------------------------------------------
# Group 1 Test
# Linear Models: OLS, Lasso & Ridge (Poly = 1), & Stepwise

for(f in 1:k) {
  train <- (id != f)
  test <- (id != f)

  matrix_poly_1 <- as.matrix(cbind(continuous_X, binary_X, dummy_x))

  # OLS
  KF_model_OLS <- lm(id_found ~ ., data = data[train,])

  # LASSO
  KF_model_poly_1_lasso_cv <- cv.glmnet(x = matrix_poly_1[train,], y = id_found[train], alpha = 1)
  KF_model_poly_1_lasso <- glmnet(x = matrix_poly_1[train,], y = id_found[train], lambda =
KF_model_poly_1_lasso_cv$lambda.min, alpha = 1)

  # Ridge
  KF_model_poly_1_ridge_cv <- cv.glmnet(x = matrix_poly_1[train,], y = id_found[train], alpha = 0)
  KF_model_poly_1_ridge <- glmnet(x = matrix_poly_1[train,], y = id_found[train], lambda =
KF_model_poly_1_ridge_cv$lambda.min, alpha = 0)

  # Stepwise
  #data_Stepwise_1 <- as.data.frame(cbind(continuous_X, binary_X, dummy_x))
```

```
  #null_1 <- lm(id_found[train] ~ 1, data = data_Stepwise_1[train,])
  #full_1 <- lm(id_found[train] ~ ., data = data_Stepwise_1[train,])

  #KF_model_forward_1 <- stepAIC(null_1, scope = list(lower = null_1, upper = full_1), trace =
FALSE, direction = 'forward')
  #KF_model_backward_1 <- stepAIC(full_1, scope = list(lower = null_1, upper = full_1), trace =
FALSE, direction = 'backward')
  #KF_model_both_1 <- stepAIC(full_1, scope = list(lower = null_1, upper = full_1), trace = FALSE,
direction = 'both')


  # Predictions
  pr_model_OLS <- predict(KF_model_OLS, newx = data[test,])
  pr_model_poly_1_lasso <- predict(KF_model_poly_1_lasso, newx = matrix_poly_1[test,])
  pr_model_poly_1_ridge <- predict(KF_model_poly_1_ridge, newx = matrix_poly_1[test,])
  #pr_model_forward_1 <- predict(KF_model_forward_1, newx = data_Stepwise_1[test,])
  #pr_model_backward_1 <- predict(KF_model_backward_1, newx = data_Stepwise_1[test,])
  #pr_model_both_1 <- predict(KF_model_both_1, newx = data_Stepwise_1[test,])

  # MSPE
  MSPE_model_OLS <- mean((pr_model_OLS - id_found[test])^2)
  MSPE_model_poly_1_lasso <- mean((pr_model_poly_1_lasso - id_found[test])^2)
  MSPE_model_poly_1_ridge <- mean((pr_model_poly_1_ridge - id_found[test])^2)
  #MSPE_model_forward_1 <- mean((pr_model_forward_1 - id_found[test])^2)
  #MSPE_model_backward_1 <- mean((pr_model_backward_1 - id_found[test])^2)
  #MSPE_model_both_1 <- mean((pr_model_both_1 - id_found[test])^2)

  print(paste('Group 1 Test - Fold', f, '/', k))
}

MSPE_df <- rbind(MSPE_df,
                 c("OLS", MSPE_model_OLS, 1),
                 c("Lasso", MSPE_model_poly_1_lasso, 1),
                 c("Ridge", MSPE_model_poly_1_ridge, 1)#,
                 #                 c("Forward Stepwise", MSPE_model_forward_1, 1),
                 #                 c("Backward Stepwise", MSPE_model_backward_1, 1),
                 #                 c("Both-Direction Stepwise", MSPE_model_both_1, 1)
)
colnames(MSPE_df) <- column_names


## ------------------------------------------------------------
# Group 2 Test
# Lasso & Ridge

degreeTest <- 3

MSPE_model_poly_x_lasso <- vector(length = degreeTest - 1)
MSPE_model_poly_x_ridge <- vector(length = degreeTest - 1)
```

```
for (deg in 2:degreeTest) {
  temp_lasso <- vector(length = k)
  temp_ridge <- vector(length = k)

  continuous_x_poly <- poly(as.matrix(continuous_X), deg)
  matrix_poly_x <- as.matrix(cbind(continuous_x_poly, binary_X, dummy_x))

  for(f in 1:k) {
    train <- (id != f)
    test <- (id != f)

    # LASSO
    KF_model_poly_x_lasso_cv <- cv.glmnet(x = matrix_poly_x[train,], y = id_found[train], alpha = 1)
    KF_model_poly_x_lasso <- glmnet(x = matrix_poly_x[train,], y = id_found[train], lambda =
KF_model_poly_x_lasso_cv$lambda.min, alpha = 1)

    # Ridge
    KF_model_poly_x_ridge_cv <- cv.glmnet(x = matrix_poly_x[train,], y = id_found[train], alpha = 0)
    KF_model_poly_x_ridge <- glmnet(x = matrix_poly_x[train,], y = id_found[train], lambda =
KF_model_poly_x_ridge_cv$lambda.min, alpha = 0)


    # Predictions
    pr_model_poly_x_lasso <- predict(KF_model_poly_x_lasso, newx = matrix_poly_x[test,])
    pr_model_poly_x_ridge <- predict(KF_model_poly_x_ridge, newx = matrix_poly_x[test,])

    # MSPE
    temp_lasso <- mean((pr_model_poly_x_lasso - id_found[test])^2)
    temp_ridge <- mean((pr_model_poly_x_ridge - id_found[test])^2)

    print(paste('Group 2 Test - Fold', f, '/', k))
  }
  MSPE_model_poly_x_lasso[deg - 1] <- min(temp_lasso)
  MSPE_model_poly_x_ridge[deg - 1] <- min(temp_ridge)

  print(paste('Group 2 Test - Degree', deg, '/', degreeTest))
}

MSPE_df <- rbind(MSPE_df,
                 c("Lasso w/ Poly", min(MSPE_model_poly_x_lasso),
which.min(MSPE_model_poly_x_lasso) + 1),
                 c("Ridge w/ Poly", min(MSPE_model_poly_x_ridge),
which.min(MSPE_model_poly_x_ridge) + 1)
)


## ------------------------------------------------------
# Group 3 Test
# High Degree Splines
```

```
degreeTest <- 100

MSPE_model_hd_spline_1 <- vector(length = degreeTest - 3)
MSPE_model_hd_spline_2 <- vector(length = degreeTest - 3)
MSPE_model_hd_spline_3 <- vector(length = degreeTest - 3)

for (deg_free in 3:degreeTest) {
 temp_spline_1 <- vector(length = k)
 temp_spline_2 <- vector(length = k)
 temp_spline_3 <- vector(length = k)

 continuous_x_poly <- poly(as.matrix(continuous_X), 2)
 matrix_poly_x <- as.matrix(cbind(continuous_x_poly, binary_X, dummy_x))

 for(f in 1:k) {
  train <- (id != f)
  test <- (id != f)

  ## High degree spline fit
  model_hd_spline_1 <- lm(id_found ~
                    bs(ev_date, df = deg_free) + . , data = data[train,]
  )

  model_hd_spline_2 <- lm(id_found ~
                    bs(address, df = deg_free) + . , data = data[train,]
  )

  model_hd_spline_3 <- lm(id_found ~
                    bs(ev_date, df = deg_free) +
                    bs(address, df = deg_free) + . , data = data[train,]
  )


  # Predictions
  pr_model_hd_spline_1 <- predict(model_hd_spline_1, newx = data[test,])
  pr_model_hd_spline_2 <- predict(model_hd_spline_2, newx = data[test,])
  pr_model_hd_spline_3 <- predict(model_hd_spline_3, newx = data[test,])

  # MSPE
  temp_spline_1 <- mean((pr_model_hd_spline_1 - id_found[test])^2)
  temp_spline_2 <- mean((pr_model_hd_spline_2 - id_found[test])^2)
  temp_spline_3 <- mean((pr_model_hd_spline_3 - id_found[test])^2)

  print(paste('Group 3 Test - Fold', f, '/', k))
 }
 MSPE_model_hd_spline_1[deg_free - 3] <- min(temp_spline_1)
 MSPE_model_hd_spline_2[deg_free - 3] <- min(temp_spline_2)
 MSPE_model_hd_spline_3[deg_free - 3] <- min(temp_spline_3)
```

```
  print(paste('Group 3 Test - Degree', deg_free, '/', degreeTest))
}

MSPE_df <- rbind(MSPE_df,
                 c("High Degree Spline Alpha", min(MSPE_model_hd_spline_1),
which.min(MSPE_model_hd_spline_1) + 1),
                 c("High Degree Spline Beta", min(MSPE_model_hd_spline_2),
which.min(MSPE_model_hd_spline_2) + 1),
                 c("High Degree Spline Charlie", min(MSPE_model_hd_spline_3),
which.min(MSPE_model_hd_spline_3) + 1)
)


## ------------------------------------------------------------
# Group 4 Test
# Stepwise

degreeTest <- 3

MSPE_model_forward_x <- vector(length = degreeTest - 1)
MSPE_model_backward_x <- vector(length = degreeTest - 1)
MSPE_model_both_x <- vector(length = degreeTest - 1)

for (deg in 2:degreeTest) {
  temp_forward <- vector(length = k)
  temp_backward <- vector(length = k)
  temp_both <- vector(length = k)

  continuous_x_poly <- poly(as.matrix(continuous_X), deg)
  data_Stepwise_x <- as.data.frame(cbind(continuous_x_poly, binary_X, dummy_x))

  for(f in 1:k) {
    train <- (id != f)
    test <- (id != f)

    # Stepwise
    null_x <- lm(id_found[train] ~ 1, data = data_Stepwise_x[train,])
    full_x <- lm(id_found[train] ~ ., data = data_Stepwise_x[train,])

    KF_model_forward_x <- stepAIC(null_x, scope = list(lower = null_x, upper = full_x), trace =
FALSE, direction = 'forward')
    KF_model_backward_x <- stepAIC(full_x, scope = list(lower = null_x, upper = full_x), trace =
FALSE, direction = 'backward')
    KF_model_both_x <- stepAIC(full_x, scope = list(lower = null_x, upper = full_x), trace = FALSE,
direction = 'both')


    # Predictions
    pr_model_forward_x <- predict(KF_model_forward_x, newx = data_Stepwise_x[test,])
    pr_model_backward_x <- predict(KF_model_backward_x, newx = data_Stepwise_x[test,])
```

```
   pr_model_both_x <- predict(KF_model_both_x, newx = data_Stepwise_x[test,])

   # MSPE
   temp_forward <- mean((pr_model_forward_x - id_found[test])^2)
   temp_backward <- mean((pr_model_backward_x - id_found[test])^2)
   temp_both <- mean((pr_model_both_x - id_found[test])^2)

   print(paste('Group 4 Test - Fold', f, '/', k))
  }
  MSPE_model_forward_x[deg - 1] <- min(temp_forward)
  MSPE_model_backward_x[deg - 1] <- min(temp_backward)
  MSPE_model_both_x[deg - 1] <- min(temp_both)

  print(paste('Group 4 Test - Degree', deg, '/', degreeTest))
}

MSPE_df <- rbind(MSPE_df,
            c("Forward Stepwise", min(MSPE_model_forward_x),
which.min(MSPE_model_forward_x) + 1),
            c("Backward Stepwise", min(MSPE_model_backward_x),
which.min(MSPE_model_backward_x) + 1),
            c("Backward Stepwise", min(MSPE_model_both_x), which.min(MSPE_model_both_x) + 1)
)


## ------------------------------------------------------
## Final Analysis
print(MSPE_df)

coefficients <- as.data.frame(as.matrix(KF_model_poly_x_lasso$beta))

plot(MSPE_model_forward_x)
plot(MSPE_model_poly_x_lasso)
plot(MSPE_model_poly_x_ridge)
plot(MSPE_model_hd_spline_1)
plot(MSPE_model_hd_spline_2)
plot(MSPE_model_hd_spline_3)


## ------------------------------------------------------
# Notebook export
wb <- createWorkbook()
addWorksheet(wb, "FI_coefficients")
writeData(wb, sheet = 1, x = coefs_df, rowNames = TRUE)
addWorksheet(wb, "MSE_Naive")
writeData(wb, sheet = 2, x = MSE_Naive, rowNames = TRUE)
addWorksheet(wb, "MSPE_df")
writeData(wb, sheet = 3, x = MSPE_df, rowNames = TRUE)

saveWorkbook(wb, "Variable Coefficients.xlsx", overwrite = TRUE)
```

APPENDIX E—NAIVE & TEST DATA TRAINING CODE

The following contains the R do-file utilized to process the data, test naive models, and train the models on test data.

```
## ------------------------------------------------------
## Library Installation
library(readxl)
library(ggplot2)
library(ggmosaic)
library(dplyr)
library(tidyr)
library(ggplot2)
library(plotly)
library(fastDummies)
library(splines)
library(glmnet)
library(MASS)
library(openxlsx)
library(boot)

## ------------------------------------------------------
## Setup
dataOrg <- read_excel("Compiled Data.xlsx", "Raw Data")
dataOrg <- dataOrg[dataOrg$id_stat != "HELP", ]
#dataOrg$reg[dataOrg$reg == "NA"] <- NA
#dataOrg$atten_type[dataOrg$atten_type == "NA"] <- NA
dataOrg <- dataOrg[!is.na(dataOrg$event_id), ]
dataOrg <- dataOrg[c("id_found", setdiff(names(dataOrg), "id_found"))]

names(dataOrg)
head(dataOrg)
summary(dataOrg)

continuous_X <- subset(dataOrg, select = c(ev_date, address))
dummy_x <- subset(dataOrg, select = c(event_id, reg, atten_type))
binary_X <- subset(dataOrg, select = c(reviewed, source_id, id_extracted, first_name, last_name,
org_name, name_title, job_title, email, phone, is_invitee, is_guest, guest_of, invite_accepted,
invite_rejected, yes_partic, no_partic, cancel_date, modif_date, action_date, academ_divis,
college_school, grad_yr, reg_date))
id_found <- dataOrg$id_found


## ------------------------------------------------------
## continuous variables

##
x <- dataOrg$address
x_label <- "# of Address Fields Provided"
```

```
y <- id_found

y <- ifelse(id_found == 1, "Matched", "Not Matched")
data_for_plot <- data.frame(x, y)
ggplot(data_for_plot, aes(x, fill = y)) +
  geom_bar(position = "fill", color = "darkgrey") +
  labs(x = x_label, y = "Density", fill = "Key") +
  scale_x_continuous(breaks = seq(0, 5, by = 1)) +
  theme_classic() +
  ggtitle(paste("Constituent match vs", x_label))


##
x <- as.Date(dataOrg$ev_date, origin = "1899-12-30")
x_label <- "Date of Event"
y <- id_found

y <- ifelse(id_found == 1, "Matched", "Not Matched")
data_for_plot <- data.frame(x, y)
ggplot(data_for_plot, aes(x, fill = y)) +
  scale_x_date(date_labels = "%Y") +
  geom_histogram(position = 'fill', color = "darkgrey", alpha = 0.5, bins = 80) +
  labs(x = x_label, y = "Density", fill = "Key") +
  theme_classic() +
  ggtitle(paste("Constituent match vs", x_label))


## ------------------------------------------------------
## dummy variables

##
x <- dataOrg$reg
x_label <- "Registration Type"
y <- id_found

y <- ifelse(id_found == 1, "Matched", "Not Matched")
data_for_plot <- data.frame(x, y)
ggplot(data_for_plot, aes(x, fill = y)) +
  geom_bar(position = "fill", color = "darkgrey") +
  labs(x = x_label, y = "Density", fill = "Key") +
  theme_classic() +
  ggtitle(paste("Constituent match vs", x_label)) +
  coord_flip()


##
x <- dataOrg$atten_type
x_label <- "Event Attendance Form"
y <- id_found
```

```
y <- ifelse(id_found == 1, "Matched", "Not Matched")
data_for_plot <- data.frame(x, y)
ggplot(data_for_plot, aes(x, fill = y)) +
  geom_bar(position = "fill", color = "darkgrey") +
  labs(x = x_label, fill = "Key") +
  theme_classic() +
  ggtitle(paste("Constituent match vs", x_label)) +
  coord_flip()


##
x <- dataOrg$event_id
y <- id_found

y <- ifelse(id_found == 1, "Matched", "Not Matched")
table_result <- as.data.frame(table(x, y))
data_for_plot <- table_result %>%
  group_by(x) %>%
  mutate(Proportion = Freq / sum(Freq)) %>%
  ungroup()

ggplot(data_for_plot, aes(x = Proportion, fill = y)) +
  geom_histogram(position = "stack", bins = 40, color = "darkgrey", alpha = 0.5) +
  labs(x = "Density", y = "# of Events", fill = "Key") +
  theme_classic() +
  ggtitle("# of Events vs Event Density between Matched & Not-Matched Constituents")


## ------------------------------------------------------------
## binary variables

# Prepare and reshape data
binary_data <- binary_X %>%
  mutate(id_found = id_found) %>%
  mutate(across(everything(), as.factor))

# Convert to long format
long_data <- binary_data %>%
  pivot_longer(cols = everything(),
               names_to = "variable",
               values_to = "value")

# Ensure 'variable' is a factor with levels in original order
long_data$variable <- factor(long_data$variable, levels = names(binary_X))

# Create Radial Plot with stylization and slice lines
ggplot(long_data, aes(x = variable, fill = value)) +
  geom_bar(width = 1, position = "fill") +
  coord_polar(start = 0) +
  geom_vline(xintercept = seq(0.5, length(unique(long_data$variable)) - 0.5, by = 1),
```

```
          color = "black", linetype = "dashed") +  # Lines for each slice
  theme_minimal(base_size = 10) +
  scale_fill_manual(values = c("0" = "yellow", "1" = "lightblue")) +
  labs(title = "Radial Binary Variable Plot",
       x = "Binary Variables",
       y = "Proportion",
       fill = "Binary Value") +
  theme(axis.text.x = element_text(angle = 0, hjust = 1),
        axis.title.x = element_text(size = 14, face = "bold"),
        axis.title.y = element_text(size = 14, face = "bold"),
        plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
        legend.position = "bottom",
        legend.title = element_text(size = 12),
        legend.text = element_text(size = 10)) +
  guides(fill = guide_legend(title = "Binary Value"))



## ------------------------------------------------------------
## DATA PROCESSING

## Drop ID_Stat
dataOrg <- subset(dataOrg, select = -id_stat)

## Date Value Converted (now number of days since a 1/1/2000)
dataOrg$ev_date <- as.numeric(as.Date(dataOrg$ev_date) - as.Date("2000-01-01"))
filtered_data <- subset(dataOrg, reviewed > 0)

# Combining Variables
#filtered_data <- filtered_data %>%
#  mutate(full_name = ifelse(first_name > 0 & last_name > 0, 1, 0))

data_with_dummies <- dummy_cols(filtered_data)

## Rare Column Removal
boolean_proportions <- colMeans(data_with_dummies == TRUE, na.rm = TRUE)
threshold <- 0.0005
low_variability_columns <- names(boolean_proportions[boolean_proportions < threshold])
low_variability_columns <- setdiff(low_variability_columns, names(continuous_X))
print(low_variability_columns)

## Remove redundant dummy var columns
cols_to_remove <- c('event_id', 'reg', 'atten_type')
cols_to_remove <- c(cols_to_remove, low_variability_columns)

data_pre_multi <- as.data.frame(data_with_dummies[, !names(data_with_dummies) %in%
cols_to_remove])

## Columns to select
dummy_x <- colnames(data_pre_multi)[grepl("^event_id_|^reg_|^atten_type_",
```

```
colnames(data_pre_multi))]
dummy_x <- dummy_x[!grepl("^reg_date$", dummy_x)]

continuous_X <- unlist(list('ev_date', 'address'))
binary_X <- unlist(list("reviewed", #"source_id", "id_extracted",
                        "first_name", "last_name", "org_name", "name_title", "job_title", "email", "phone",
"is_invitee", "is_guest", "guest_of", "invite_accepted", "invite_rejected", "yes_partic", "no_partic",
"cancel_date", "modif_date", "action_date", "academ_divis", "college_school", "grad_yr", "reg_date"))

## Remove multi-collinearity
OLS <- lm(id_found ~ ., data = data_pre_multi)
summary(OLS)

keep_var <- names(OLS$coefficients)[!is.na(OLS$coefficients)][-1]
keep <- c('id_found', keep_var)
multi_colin_vars <- gsub("`", "", keep)
#multi_colin_vars <- setdiff(keep, binary_vars_to_remove)
data <- subset(data_pre_multi, select = multi_colin_vars)

## Subset Creation
dummy_x <- intersect(names(data), dummy_x)
continuous_X <- intersect(names(data), continuous_X)
binary_X <- intersect(names(data), binary_X)

dummy_x <- subset(data, select = dummy_x)
continuous_X <- subset(data, select = continuous_X)
binary_X <- subset(data, select = binary_X)
id_found <- data$id_found
data <- cbind(id_found, binary_X, continuous_X, dummy_x)

removed_var <- setdiff(names(data_with_dummies), multi_colin_vars)
print(removed_var)

#print(any(is.na(data)))
#print(sapply(data, function(x) sum(is.na(x))))
#print(unique(filtered_data$atten_type))


## ------------------------------------------------------------
## Simple Model: OLS with all provided variables
model_OLS <- lm(id_found ~ ., data = data)
plot(model_OLS)
summary(model_OLS)


## ------------------------------------------------------------
## LASSO & RIDGE REGRESSION MODELS

## Polynomial Matrix Creation
continuous_2_poly <- poly(as.matrix(continuous_X), 2)
```

```
matrix_poly_1 <- as.matrix(cbind(continuous_X, binary_X, dummy_x))
matrix_poly_2 <- as.matrix(cbind(continuous_2_poly, binary_X, dummy_x))

set.seed(6123)
# LASSO Regressions:
model_poly_1_lasso_cv <- cv.glmnet(x = matrix_poly_1, y = id_found, alpha = 1)
model_poly_1_lasso <- glmnet(x = matrix_poly_1, y = id_found, lambda =
model_poly_1_lasso_cv$lambda.min, alpha = 1)

model_poly_2_lasso_cv <- cv.glmnet(x = matrix_poly_2, y = id_found, alpha = 1)
model_poly_2_lasso <- glmnet(x = matrix_poly_2, y = id_found, lambda =
model_poly_2_lasso_cv$lambda.min, alpha = 1)

# Check betas
model_poly_1_lasso$beta
model_poly_2_lasso$beta
plot(model_poly_1_lasso_cv)
plot(model_poly_1_lasso_cv$glmnet.fit)
plot(model_poly_2_lasso_cv)
plot(model_poly_2_lasso_cv$glmnet.fit)


# Ridge Regressions:
model_poly_1_ridge_cv <- cv.glmnet(x = matrix_poly_1, y = id_found, alpha = 0)
model_poly_1_ridge <- glmnet(x = matrix_poly_1, y = id_found, lambda =
model_poly_1_ridge_cv$lambda.min, alpha = 0)

model_poly_2_ridge_cv <- cv.glmnet(x = matrix_poly_2, y = id_found, alpha = 0)
model_poly_2_ridge <- glmnet(x = matrix_poly_2, y = id_found, lambda =
model_poly_2_ridge_cv$lambda.min, alpha = 0)

# Check betas
model_poly_1_ridge$beta
model_poly_2_ridge$beta
plot(model_poly_1_ridge_cv)
plot(model_poly_1_ridge_cv$glmnet.fit)
plot(model_poly_2_ridge_cv)
plot(model_poly_2_ridge_cv$glmnet.fit)


## ------------------------------------------------------
## High degree spline fit
df <- 3
model_hd_spline_1 <- lm(id_found ~
                          bs(ev_date, df = df) + . , data = data
                        )

df <- 3
model_hd_spline_2 <- lm(id_found ~
```

```
                    bs(address, df = df) + . , data = data
                  )

df <- 3
model_hd_spline_3 <- lm(id_found ~
                    bs(ev_date, df = df) +
                    bs(address, df = df) + . , data = data
                  )

summary(model_hd_spline_1)
plot(model_hd_spline_1)
plot(model_hd_spline_2)
plot(model_hd_spline_3)


## ------------------------------------------------------
## Stepwise Model
data_Stepwise_1 <- as.data.frame(cbind(continuous_X, binary_X, dummy_x))

null_1 <- lm(id_found ~ 1, data = data_Stepwise_1)
full_1 <- lm(id_found ~ ., data = data_Stepwise_1)

model_forward_1 <- stepAIC(null_1, scope = list(lower = null_1, upper = full_1), trace = FALSE,
direction = 'forward')
model_backward_1 <- stepAIC(full_1, scope = list(lower = null_1, upper = full_1), trace = FALSE,
direction = 'backward')
model_both_1 <- stepAIC(full_1, scope = list(lower = null_1, upper = full_1), trace = FALSE,
direction = 'both')


data_Stepwise_2 <- as.data.frame(cbind(continuous_2_poly, binary_X, dummy_x))

null_2 <- lm(id_found ~ 1, data = data_Stepwise_2)
full_2 <- lm(id_found ~ ., data = data_Stepwise_2)

model_forward_2 <- stepAIC(null_2, scope = list(lower = null_2, upper = full_2), trace = FALSE,
direction = 'forward')
model_backward_2 <- stepAIC(full_2, scope = list(lower = null_2, upper = full_2), trace = FALSE,
direction = 'backward')
model_both_2 <- stepAIC(full_2, scope = list(lower = null_2, upper = full_2), trace = FALSE,
direction = 'both')


plot(model_forward_1)
plot(model_backward_1)
plot(model_forward_2)
plot(model_backward_2)


## ------------------------------------------------------
```

```
#MSE Calculation Naive

#Fitting models:
calculate_MSE_linear <- function(model) {
  residuals <- model$residuals
  mse <- mean(residuals^2)
  return(mse)
}

calculate_MSE_complex <- function(model, matrix, dependent_variable) {
  residuals <- predict(model, newx = matrix)
  mse <- mean((residuals - dependent_variable)^2)
  return(mse)
}

column_names <- c("Model Name", "MSE", "Polynomial Degree or Degress of Freedom")
MSE_Naive <- data.frame(matrix(ncol = length(column_names), nrow = 0))
MSE_Naive <- rbind(MSE_Naive,
                   c("model_OLS", calculate_MSE_linear(model_OLS), 1),
                   c("model_hd_spline_1", calculate_MSE_linear(model_hd_spline_1), 3),
                   c("model_hd_spline_2", calculate_MSE_linear(model_hd_spline_2), 3),
                   c("model_hd_spline_3", calculate_MSE_linear(model_hd_spline_3), 3),
                   c("model_forward_1", calculate_MSE_linear(model_forward_1), 1),
                   c("model_backward_1", calculate_MSE_linear(model_backward_1), 1),
                   c("model_both_1", calculate_MSE_linear(model_both_1), 1),
                   c("model_forward_2", calculate_MSE_linear(model_forward_2), 2),
                   c("model_backward_2", calculate_MSE_linear(model_backward_2), 2),
                   c("model_both_2", calculate_MSE_linear(model_both_2), 2),
                   c("model_poly_1_ridge", calculate_MSE_complex(model_poly_1_ridge, matrix_poly_1,
id_found), 1),
                   c("model_poly_2_ridge", calculate_MSE_complex(model_poly_2_ridge, matrix_poly_2,
id_found), 2),
                   c("model_poly_1_lasso", calculate_MSE_complex(model_poly_1_lasso, matrix_poly_1,
id_found), 1),
                   c("model_poly_2_lasso", calculate_MSE_complex(model_poly_2_lasso, matrix_poly_2,
id_found), 2)
)
colnames(MSE_Naive) <- column_names
print(MSE_Naive)


## ------------------------------------------------------
summary(model_hd_spline_3)
model_poly_2_lasso$beta

plot(model_OLS, title="OLS Model")
plot(model_poly_2_ridge_cv$glmnet.fit, title="Lasso (deg 2) Model")
plot(model_poly_2_lasso_cv$glmnet.fit, title="Ridge (deg 2) Model")
plot(model_hd_spline_3, title="High Degree Spline 3 Model")
plot(model_both_2, title="Stepwise Both (deg 2) Model")
```

```
## ------------------------------------------------------------
### Compare models via K-fold Cross Validation
k <- 10
n <- length(id_found)

set.seed(7142)
id <- sample(rep(1:k, length = n))

column_names <- c("Model Name", "MSPE", "Polynomial Degree or Degress of Freedom")
MSPE_df <- data.frame(matrix(ncol = length(column_names), nrow = 0))
colnames(MSPE_df) <- column_names

## ------------------------------------------------------------
# Group 1 Test
# Linear Models: OLS, Lasso & Ridge (Poly = 1), & Stepwise

for(f in 1:k) {
  train <- (id != f)
  test <- (id == f)

  matrix_poly_1 <- as.matrix(cbind(continuous_X, binary_X, dummy_x))

  # OLS
  KF_model_OLS <- lm(id_found ~ ., data = data[train,])

  # LASSO
  KF_model_poly_1_lasso_cv <- cv.glmnet(x = matrix_poly_1[train,], y = id_found[train], alpha = 1)
  KF_model_poly_1_lasso <- glmnet(x = matrix_poly_1[train,], y = id_found[train], lambda =
KF_model_poly_1_lasso_cv$lambda.min, alpha = 1)

  # Ridge
  KF_model_poly_1_ridge_cv <- cv.glmnet(x = matrix_poly_1[train,], y = id_found[train], alpha = 0)
  KF_model_poly_1_ridge <- glmnet(x = matrix_poly_1[train,], y = id_found[train], lambda =
KF_model_poly_1_ridge_cv$lambda.min, alpha = 0)

  # Stepwise
  data_Stepwise_1 <- as.data.frame(cbind(continuous_X, binary_X, dummy_x))

  null_1 <- lm(id_found[train] ~ 1, data = data_Stepwise_1[train,])
  full_1 <- lm(id_found[train] ~ ., data = data_Stepwise_1[train,])

  KF_model_forward_1 <- stepAIC(null_1, scope = list(lower = null_1, upper = full_1), trace =
FALSE, direction = 'forward')
  KF_model_backward_1 <- stepAIC(full_1, scope = list(lower = null_1, upper = full_1), trace =
FALSE, direction = 'backward')
  KF_model_both_1 <- stepAIC(full_1, scope = list(lower = null_1, upper = full_1), trace = FALSE,
direction = 'both')
```

```
  # Predictions
  pr_model_OLS <- predict(KF_model_OLS, newx = data[test,])
  pr_model_poly_1_lasso <- predict(KF_model_poly_1_lasso, newx = matrix_poly_1[test,])
  pr_model_poly_1_ridge <- predict(KF_model_poly_1_ridge, newx = matrix_poly_1[test,])
  pr_model_forward_1 <- predict(KF_model_forward_1, newx = data_Stepwise_1[test,])
  pr_model_backward_1 <- predict(KF_model_backward_1, newx = data_Stepwise_1[test,])
  pr_model_both_1 <- predict(KF_model_both_1, newx = data_Stepwise_1[test,])

  # MSPE
  MSPE_model_OLS <- mean((pr_model_OLS - id_found[test])^2)
  MSPE_model_poly_1_lasso <- mean((pr_model_poly_1_lasso - id_found[test])^2)
  MSPE_model_poly_1_ridge <- mean((pr_model_poly_1_ridge - id_found[test])^2)
  MSPE_model_forward_1 <- mean((pr_model_forward_1 - id_found[test])^2)
  MSPE_model_backward_1 <- mean((pr_model_backward_1 - id_found[test])^2)
  MSPE_model_both_1 <- mean((pr_model_both_1 - id_found[test])^2)

  print(paste('Group 1 Test - Fold', f, '/', k))
}

MSPE_df <- rbind(MSPE_df,
                 c("OLS", MSPE_model_OLS, 1),
                 c("Lasso", MSPE_model_poly_1_lasso, 1),
                 c("Ridge", MSPE_model_poly_1_ridge, 1),
                 c("Forward Stepwise", MSPE_model_forward_1, 1),
                 c("Backward Stepwise", MSPE_model_backward_1, 1),
                 c("Both-Direction Stepwise", MSPE_model_both_1, 1)
                 )
colnames(MSPE_df) <- column_names


## ------------------------------------------------------
# Group 2 Test
# Lasso & Ridge

degreeTest <- 3

MSPE_model_poly_x_lasso <- vector(length = degreeTest - 1)
MSPE_model_poly_x_ridge <- vector(length = degreeTest - 1)

for (deg in 2:degreeTest) {
  temp_lasso <- vector(length = k)
  temp_ridge <- vector(length = k)

  continuous_x_poly <- poly(as.matrix(continuous_X), deg)
  matrix_poly_x <- as.matrix(cbind(continuous_x_poly, binary_X, dummy_x))

  for(f in 1:k) {
    train <- (id != f)
    test <- (id == f)
```

```
   # LASSO
   KF_model_poly_x_lasso_cv <- cv.glmnet(x = matrix_poly_x[train,], y = id_found[train], alpha = 1)
   KF_model_poly_x_lasso <- glmnet(x = matrix_poly_x[train,], y = id_found[train], lambda =
KF_model_poly_x_lasso_cv$lambda.min, alpha = 1)

   # Ridge
   KF_model_poly_x_ridge_cv <- cv.glmnet(x = matrix_poly_x[train,], y = id_found[train], alpha = 0)
   KF_model_poly_x_ridge <- glmnet(x = matrix_poly_x[train,], y = id_found[train], lambda =
KF_model_poly_x_ridge_cv$lambda.min, alpha = 0)


   # Predictions
   pr_model_poly_x_lasso <- predict(KF_model_poly_x_lasso, newx = matrix_poly_x[test,])
   pr_model_poly_x_ridge <- predict(KF_model_poly_x_ridge, newx = matrix_poly_x[test,])

   # MSPE
   temp_lasso <- mean((pr_model_poly_x_lasso - id_found[test])^2)
   temp_ridge <- mean((pr_model_poly_x_ridge - id_found[test])^2)

   print(paste('Group 2 Test - Fold', f, '/', k))
  }
  MSPE_model_poly_x_lasso[deg - 1] <- min(temp_lasso)
  MSPE_model_poly_x_ridge[deg - 1] <- min(temp_ridge)

  print(paste('Group 2 Test - Degree', deg, '/', degreeTest))
}

MSPE_df <- rbind(MSPE_df,
                 c("Lasso w/ Poly", min(MSPE_model_poly_x_lasso),
which.min(MSPE_model_poly_x_lasso) + 1),
                 c("Ridge w/ Poly", min(MSPE_model_poly_x_ridge),
which.min(MSPE_model_poly_x_ridge) + 1)
)


## ------------------------------------------------------------
# Group 3 Test
# High Degree Splines

degreeTest <- 100

MSPE_model_hd_spline_1 <- vector(length = degreeTest - 3)
MSPE_model_hd_spline_2 <- vector(length = degreeTest - 3)
MSPE_model_hd_spline_3 <- vector(length = degreeTest - 3)

for (deg_free in 3:degreeTest) {
  temp_spline_1 <- vector(length = k)
  temp_spline_2 <- vector(length = k)
  temp_spline_3 <- vector(length = k)
```

```
 continuous_x_poly <- poly(as.matrix(continuous_X), 2)
 matrix_poly_x <- as.matrix(cbind(continuous_x_poly, binary_X, dummy_x))

 for(f in 1:k) {
  train <- (id != f)
  test <- (id == f)

  ## High degree spline fit
  model_hd_spline_1 <- lm(id_found ~
                    bs(ev_date, df = deg_free) + . , data = data[train,]
  )

  model_hd_spline_2 <- lm(id_found ~
                    bs(address, df = deg_free) + . , data = data[train,]
  )

  model_hd_spline_3 <- lm(id_found ~
                    bs(ev_date, df = deg_free) +
                    bs(address, df = deg_free) + . , data = data[train,]
  )


  # Predictions
  pr_model_hd_spline_1 <- predict(model_hd_spline_1, newx = data[test,])
  pr_model_hd_spline_2 <- predict(model_hd_spline_2, newx = data[test,])
  pr_model_hd_spline_3 <- predict(model_hd_spline_3, newx = data[test,])

  # MSPE
  temp_spline_1 <- mean((pr_model_hd_spline_1 - id_found[test])^2)
  temp_spline_2 <- mean((pr_model_hd_spline_2 - id_found[test])^2)
  temp_spline_3 <- mean((pr_model_hd_spline_3 - id_found[test])^2)

  print(paste('Group 3 Test - Fold', f, '/', k))
 }
 MSPE_model_hd_spline_1[deg_free - 3] <- min(temp_spline_1)
 MSPE_model_hd_spline_2[deg_free - 3] <- min(temp_spline_2)
 MSPE_model_hd_spline_3[deg_free - 3] <- min(temp_spline_3)

 print(paste('Group 3 Test - Degree', deg_free, '/', degreeTest))
}

MSPE_df <- rbind(MSPE_df,
            c("High Degree Spline Alpha", min(MSPE_model_hd_spline_1),
which.min(MSPE_model_hd_spline_1) + 1),
            c("High Degree Spline Beta", min(MSPE_model_hd_spline_2),
which.min(MSPE_model_hd_spline_2) + 1),
            c("High Degree Spline Charlie", min(MSPE_model_hd_spline_3),
which.min(MSPE_model_hd_spline_3) + 1)
)
```

```
## ------------------------------------------------------------
# Group 4 Test
# Stepwise

degreeTest <- 3

MSPE_model_forward_x <- vector(length = degreeTest - 1)
MSPE_model_backward_x <- vector(length = degreeTest - 1)
MSPE_model_both_x <- vector(length = degreeTest - 1)

for (deg in 2:degreeTest) {
  temp_forward <- vector(length = k)
  temp_backward <- vector(length = k)
  temp_both <- vector(length = k)

  continuous_x_poly <- poly(as.matrix(continuous_X), deg)
  data_Stepwise_x <- as.data.frame(cbind(continuous_x_poly, binary_X, dummy_x))

  for(f in 1:k) {
    train <- (id != f)
    test <- (id == f)

    # Stepwise
    null_x <- lm(id_found[train] ~ 1, data = data_Stepwise_x[train,])
    full_x <- lm(id_found[train] ~ ., data = data_Stepwise_x[train,])

    KF_model_forward_x <- stepAIC(null_x, scope = list(lower = null_x, upper = full_x), trace =
FALSE, direction = 'forward')
    KF_model_backward_x <- stepAIC(full_x, scope = list(lower = null_x, upper = full_x), trace =
FALSE, direction = 'backward')
    KF_model_both_x <- stepAIC(full_x, scope = list(lower = null_x, upper = full_x), trace = FALSE,
direction = 'both')


    # Predictions
    pr_model_forward_x <- predict(KF_model_forward_x, newx = data_Stepwise_x[test,])
    pr_model_backward_x <- predict(KF_model_backward_x, newx = data_Stepwise_x[test,])
    pr_model_both_x <- predict(KF_model_both_x, newx = data_Stepwise_x[test,])

    # MSPE
    temp_forward <- mean((pr_model_forward_x - id_found[test])^2)
    temp_backward <- mean((pr_model_backward_x - id_found[test])^2)
    temp_both <- mean((pr_model_both_x - id_found[test])^2)

    print(paste('Group 4 Test - Fold', f, '/', k))
  }
  MSPE_model_forward_x[deg - 1] <- min(temp_forward)
  MSPE_model_backward_x[deg - 1] <- min(temp_backward)
  MSPE_model_both_x[deg - 1] <- min(temp_both)
```

```
  print(paste('Group 4 Test - Degree', deg, '/', degreeTest))
}

MSPE_df <- rbind(MSPE_df,
                 c("Forward Stepwise", min(MSPE_model_forward_x),
which.min(MSPE_model_forward_x) + 1),
                 c("Backward Stepwise", min(MSPE_model_backward_x),
which.min(MSPE_model_backward_x) + 1),
                 c("Backward Stepwise", min(MSPE_model_both_x), which.min(MSPE_model_both_x) + 1)
                 )


## ------------------------------------------------------------
## Final Analysis
print(MSPE_df)

coefficients <- as.data.frame(as.matrix(KF_model_poly_x_lasso$beta))

plot(KF_model_OLS)
plot(KF_model_poly_x_lasso_cv$glmnet.fit)
plot(KF_model_both_1)
plot(model_hd_spline_3)

plot(MSPE_model_hd_spline_1)
plot(MSPE_model_hd_spline_2)
plot(MSPE_model_hd_spline_3)

## ------------------------------------------------------------
## Chosen Model
## Lasso Polynomial Degree 3
k <- 20
n <- length(id_found)

set.seed(8888)
id <- sample(rep(1:k, length = n))

deg <- 3

continuous_x_poly <- poly(as.matrix(continuous_X), deg)
matrix_poly_x <- as.matrix(cbind(continuous_x_poly, binary_X, dummy_x))

for(f in 1:k) {
  print(paste("Phase", f, "0%"))
  train <- (id != f)
  test <- (id == f)

  print(paste("Phase", f, "29%"))
  # LASSO
  FI_model_poly_x_lasso_cv <- cv.glmnet(x = matrix_poly_x[train,], y = id_found[train], alpha = 1)
```

```
  FI_model_poly_x_lasso <- glmnet(x = matrix_poly_x[train,], y = id_found[train], lambda =
FI_model_poly_x_lasso_cv$lambda.min, alpha = 1)

  print(paste("Phase", f, "57%"))
  # Predictions
  pr_model_FI <- predict(FI_model_poly_x_lasso, newx = matrix_poly_x[test,])
  MSPE_plot_data <- (pr_model_FI - id_found[test])^2

  print(paste("Phase", f, "86%"))
  # MSPE
  MSPE_FI <- mean((pr_model_FI - id_found[test])^2)
  print(paste("Phase", f, "100%"))
}


## ------------------------------------------------------
#Statistical Significance

boot_lasso <- function(data, indices) {
  # Subset data based on bootstrap indices
  boot_data <- data[indices, ]

  continuous_x_poly <- poly(as.matrix(boot_data[, names(continuous_X)]), deg = 3)
  matrix_poly_x <- as.matrix(cbind(continuous_x_poly, boot_data[, names(binary_X)], boot_data[,
names(dummy_x)]))

  lasso_cv <- cv.glmnet(x = matrix_poly_x, y = boot_data$id_found, alpha = 1)
  lasso_model <- glmnet(x = matrix_poly_x, y = boot_data$id_found, lambda = lasso_cv$lambda.min,
alpha = 1)

  return(as.vector(coef(lasso_model)))
}

n_iterations <- 10000

boot_results <- boot(data = data, statistic = boot_lasso, R = n_iterations)

# Calculating Stat Sig
boot_coefs <- boot_results$t[, -1]
coefs_ci <- t(apply(boot_coefs, 2, quantile, probs = c(0.025, 0.975)))

coefs <- as.matrix(FI_model_poly_x_lasso$beta)
significant <- ifelse((coefs_ci[, 1] <= coefs) * (coefs_ci[, 2] >= coefs), TRUE, FALSE)

coefs_df <- data.frame(Coefficient = rownames(coefs),
                       Estimate = coefs,
                       Lower_CI = coefs_ci[, 1],
                       Upper_CI = coefs_ci[, 2],
                       Significant = significant)
```

```
## ------------------------------------------------------------
# FI Charts

plot(FI_model_poly_x_lasso_cv)
plot(FI_model_poly_x_lasso_cv$glmnet.fit)
plot(MSPE_plot_data)
length(MSPE_plot_data[MSPE_plot_data > 0.1]) / length(MSPE_plot_data)
length(MSPE_plot_data[MSPE_plot_data > 0.05]) / length(MSPE_plot_data)
plot(pr_model_FI)
coef(FI_model_poly_x_lasso_cv, s = "lambda.min")

FI_coefficients <- as.data.frame(as.matrix(FI_model_poly_x_lasso$beta))
FI_coefficients$s0 <- format(FI_coefficients$s0, scientific = FALSE, digits = 10)


## ------------------------------------------------------------
# Notebook export
wb <- createWorkbook()
addWorksheet(wb, "FI_coefficients")
writeData(wb, sheet = 1, x = coefs_df, rowNames = TRUE)
addWorksheet(wb, "MSE_Naive")
writeData(wb, sheet = 2, x = MSE_Naive, rowNames = TRUE)
addWorksheet(wb, "MSPE_df")
writeData(wb, sheet = 3, x = MSPE_df, rowNames = TRUE)

saveWorkbook(wb, "Variable Coefficients.xlsx", overwrite = TRUE)
```

This page intentionally left blank.