W. P. CAREY
SCHOOL of BUSINESS
ARIZONA STATE UNIVERSITY

W. P. CAREY SCHOOL OF BUSINESS HONOR CODE MASTER'S PROGRAMS

We, the students of the W.P. Carey Master's programs, are committed to maintaining the highest levels of honesty, integrity, and professional courtesy for ourselves and amongst our classmates. We believe that ethical behavior is not only the cornerstone of a good academic program, but of good business as well. As such, we are steadfast in our commitment to the following principles:

- Academic Integrity – We view the W.P. Carey Academic Integrity Policy as a living, breathing document, not just concepts on a page. As such, we are resolute in ensuring that we abide by the principles contained within, both in letter and in spirit. Furthermore, we will hold our classmates to the same high standards to which we hold ourselves.

- Consideration of Others – As W. P. Carey students, we believe that we should treat others as we wish to be treated. To this end, we will be considerate of classmates, professors, and recruiters in all aspects. Furthermore, we will constructively add to the collaborative environment by helping and encouraging others.

- Professionalism – As W. P. Carey students, we believe that the behaviors we practice today will become habit in the future. To that end, we are committed to act in a professional manner in all school settings, including classes, networking events, and interviews. We will be cognizant of the fact that we are representing the program and our classmates at all times.

Full Name (printed): Jeff Luczak

Signature Jeff L Date 11/28/2020

Full Name (printed): Devon Ramos

Signature Devon Date 11/28/2020

Full Name (printed): Howard Cho

Signature Howard Cho Date 11/28/2020

Online MSBA Applied Project Report
W.P. Carey, ASU

Topic	Life Insurance: Cycle Time and Placement Rate Analysis
Team	Team 10
Team Members	Howard Cho, Jeff Luczak, Devon Ramos
Client information	Edward Jones, Insurance Analysis & Case Oversight Devon Ramos, Business Support Specialist II

Executive Summary

Edward Jones is looking to set fair service level agreements with its life insurance carrier partners. Historically, Edward Jones' life insurance carriers take an average of two to three months to complete a client's application. Edward Jones wants to find out if this duration is an adequate SLA for all six of its life insurance carriers, or if there should be adjustments to better align with current market conditions.

To attract more business towards its life insurance unit, Edward Jones needs to provide consistency to all its clients, regardless of what products and services they have purchased. Since Edward Jones does not sell life insurance by itself, it must partner with life insurance carriers to provide the product to its clients and provide the same level of customer expectation regardless of which life insurance carrier the client ultimately chooses.

Using Python and data collected from Edward Jones and the European CDC, we built a machine learning model to predict cycle times given a new policy. Due to the abundance of variance in the data, we tried two approaches. The first approach was a model that predicted cycle time for all carriers together. The second model independently looked at each carrier. Approach two outperformed our first model when evaluated by R^2.

When comparing our predictive model to the actual values we noticed our model predicted our average SLA for full underwriting to be 53 days and express underwriting to be 34 days. These were both about 5 days faster than the actual averages. We believe the gap came from the variability in our dataset due to each carriers' process differences. To help explain this gap, our model would benefit from added features from Edward Jones. Some features that could assist would be knowing what vendors the insurance provider is using to administer the exams, interviews, etc., if a case has an associated application, and what the premium is for the policy. Adding additional features may assist with explaining the variability better and improving our model output. As it currently stands the highest correlation between one feature and our target variable is around 38%.

Our model was also able to show us the impact of both APS and 1035 exchanges on cycle time. Both elements added over 20 days to the time of the application when they were present.

The recommendation would be that an overall cycle time SLA would be approximately 60 days. To stay on target with this time period, focus needs to be spent on the underwriting stage which has the longest cycle time of any stage in the process and the most variability.

Background

Edward Jones serves nearly seven-million investors from more offices than any other investment firm in America. To best serve their clients with insurance products, Edward Jones uses set standards from Service Level Agreements (SLAs) to hold insurance carriers accountable. The SLAs are also used to set a realistic expectation for clients in terms of how long the insurance application process will take, as well as the likelihood of being approved.

Historically, an Edward Jones client waits an average of two to three months to receive their life insurance policy after initial submission of an application. Each case progresses through a series of "stages" before reaching a final outcome of policy placement (inforce). The stages are as follows: Submitted, Underwriting, Approval, Issued, Inforce. As the case progresses, there are multiple requirements that an insurance carrier may order that can cause delays such as Paramedical Exams, Attending Physician Statements (APS) & 1035 Exchanges. These requirements often require external vendors to get in contact with the insured to set up an appointment or collect information. Paramedical exams are generally quicker to complete than APS and 1035 Exchanges. APS and 1035 Exchanges typically take up the bulk of the time during the insurance application process.

Paramedical exams ordered for the insured during the life insurance application process require the insured to schedule their own appointments with a mobile nurse or with a nurse at a clinic. Paramedical exams are very similar to annual physical exams. The nurse measures the client's current height, weight, vitals, and the client provides blood and urine samples to a lab for analysis for any abnormal health readings such as high cholesterol and/or high blood sugar levels. These exams are typically quicker to complete compared with APS and 1035 Exchanges because the exams happen relatively early during the application process, and the insured is more inclined to complete exams quickly since they just invested a significant amount of time to apply for life insurance. In addition, most of the control lies with the insured during the paramedical exam. Without completion of the exams the application cannot move forward. Therefore, it is in the insured's best interest to complete the exams quickly.

When an APS is ordered for an insured's case the insured must obtain the APS from his/her personal physician. An APS is similar to a doctor's note, declaring the insured's current and historical health condition. The APS process duration could increase significantly during an application due to how proactive the insured and the insured's physician(s) are in completing the APS. Physicians are not held to the same SLAs that Edward Jones' life insurance partners are held to, which then contributes to the variability in the data. It is also worth noting that multiple APS may be ordered to address specific medical procedures completed or an insured that has multiple physicians.

A 1035 Exchange is a provision in the Internal Revenue Service code which is performed when the client wants to transition from one policy to another. Out of the three types of life insurance policies, the permanent and hybrid life insurance policies provide the client with an investment account which can increase in value. In contrast, term life insurance policies do not have the investment account component, as the client is essentially paying a subscription to have the benefit of a life insurance payout in the event the client's death occurs. 1035 Exchanges benefit the client during policy exchanges because the cash values of the existing policy would be transferred without tax penalties. Some scenarios of policy transitions do not involve 1035 Exchanges, mainly due to the lack of cash value from the originating policy. For example, a term life insurance policy does not have a cash value because it does not have an investment account component, therefore a term policy transfer would never have a 1035 Exchange associated with it.

Besides the most common information requests and tax due diligence outlined above, the insured's age, face amount of the policy and risk class can all attribute to the need for additional information collected by the insurance carrier, further delaying the process. As a result, each client's life insurance policy application process can vary significantly, which becomes a major factor that contributes to the variance in the processing days.

While there are similarities, each carrier has their own processes and work with multiple vendors to collect data on the proposed insured. For example, five of the six carriers conduct the 1035 exchange in the "approval" stage whereas the sixth carrier conducts theirs in the "underwriting stage". Additionally, some carriers offer specific products that are ineligible for express underwriting, APS, exams, etc. These unique aspects of the carrier processes must be considered when determining the best approach to determine fair SLAs.

Problem Statement

Edward Jones' leadership in the Insurance Area has asked for an analysis of data related to the submission and placement of life insurance policies over the last 12 months. The findings from this analysis will be used to reaffirm existing SLAs with their insurance carriers or to build a case for adjustments to better align with the current market conditions.

A thorough analysis of the data is required to answer the following questions:

1. What factors have a significant/insignificant influence on cycle time?
2. What is an appropriate cycle time SLA for each of the following categories?
    a. Days in Submitted
    b. Days in Underwriting
    c. Days in Approval
    d. Days in Issued
    e. Days Submitted to Final (Inforce policies only)
    f. Express Underwriting vs. Full Underwriting
    g. Inclusion of an APS(s) and/or 1035 exchange
    h. Inclusion of an Interview and/or Exam
3. What is an appropriate Placement Rate SLA for each of the following categories?
    a. Three-Month Placement Rate
    b. Six-Month Placement Rate
    c. Overall
4. Are the SLA expectations from #2 and #3 consistent with the population prior to March and post-March (COVID-19 implications)?
5. Is there significant value in splitting the SLAs by other factors that influence CT and PR (i.e. split by product)?

Edward Jones currently partners with six life insurance carriers to provide life insurance products to its clients. From a business standpoint, Edward Jones strives to provide the same high-quality service to all its clients regardless of which carrier the client selects to move forward with and apply. Clients generally do not know the details behind what makes each carrier different from one another, but each life insurance quote received from the carriers are presented the same way, allowing for an "apples to apples" comparison. The clients generally select the life insurance carrier based on recommendations from the Edward Jones advisor, and generally the carriers that have the best reputation increase their chances of getting business from the client.

As life insurance is far from the top-selling product at Edward Jones, most life insurance applications are a result of the Edward Jones advisors cross-selling other product offerings within Edward Jones. Therefore, it is important for Edward Jones to deliver all life insurance products in a consistent, high-quality manner to its clients. Edward Jones aims to provide a seamless experience for its clients across all its products, so it works with its life insurance carrier partners to drive performance towards agreed upon SLAs for each step of the life insurance application process. These SLAs will give the clients a general idea of what duration to expect during each step of the application, and it gives Edward Jones a baseline to monitor for any applications that may be taking too long, which might affect the client's satisfaction.

Edward Jones' leadership for the life insurance products area will be the main audience for our analysis. The leadership will be able to read through our analysis steps and machine learning models to reach our final recommendation. By downloading Jupyter, the leadership team can run our machine learning model to replicate our results.

The data provided is assumed to be representative of recent cycle time trends. Therefore, further analysis of data from previous years would yield little benefits. It is also noteworthy that the data for applications submitted after March 2020 are most indicative of the current working environment and book of business. After the coronavirus was classified as a pandemic by the World Health Organization on March 11, 2020, Edward Jones has made changes to the life insurance product options available to clients, as well as advancements in electronic capabilities to better serve in a remote-working environment.

Methods

The ensuing section walks through how we set up our machine learning model to predict service level times. It required some preprocessing prior to creating a model. These steps are defined below.

Overall Approach

Our approach to the problem of predicting SLAs was to first understand the correlation between each feature provided in our dataset with our target feature. The first model we built was to be able to predict the entire cycle time of a life insurance application. This feature was "Days Submitted to Final." Seeing how we had six different companies included in our dataset we had to decide whether to build a predictive model for each individual carrier or one model for all. We chose to do both as Edward Jones will not allow certain carriers to take longer on their SLA than others and reward poor cycle times. The reason we segmented our model down to each carrier was in hopes of improving our prediction accuracy. Each carrier has nuances on why their process is different and we feared our overall model was not taking that into consideration. Once we had our predictions, we looked at exploring all the steps in the life insurance application approval process to understand the average time it takes in each step.

We also attempted to gather more data to include in our model. Since we already had all the information we could gather from Edward Jones on these applications, we looked to bring in outside data. For this we chose to gather COVID cases in the United States. Our thought was as COVID cases rose we may see an increase in cycle times due to workers being furloughed and stay at home orders in place. This would make getting an APS or Exam more time consuming in theory.

We were lucky to have received the dataset from Edward Jones in a timely manner, which allowed us to start the model building process early. Our goal was to build a predictive model using the remaining 50,956 rows of filtered data to determine the most likely number of days from application submitted until the policy is final and inforce. Our model will also be used to make a case for reasonable SLAs given the current performance of all the carriers collectively. The machine learning theories that we learned during our MSBA program gave us the foundations to develop a predictive model. We selected the Python programming language and its powerful open-sourced Pandas and scikit-learn libraries to take on this problem due to our familiarity with Python and its industry-wide adoption for machine learning applications.

Exploratory Data Analysis

Edward Jones provided data for 87,672 submitted insurance applications from 9/1/2019 through 8/31/2020. The data includes separate fields for each of the variables listed in the Problem Statement and was filtered on the "Stage" column for values of only "Final - Inforce". 50,956 rows remain because of the filter, which are the placed policies that will be the primary focus of the project to determine cycle time SLAs. This filter already addressed one of the significant risks we identified: negative values in the dataset. The negative values found were related to applications in all stages apart from "Final-Inforce", so our proposed model would be unaffected. The remaining data was utilized with placement rate calculations, but the negatives would not have an effect due to the nature of the calculated fields detailed later in the report.

Before going about handling the nuances in our dataset and building a model, we needed to better understand our data. We started by checking for seasonality in our data to view how COVID may have impacted the business. Starting with September 2019 and going until the last set of applications in our dataset (August 2020), we viewed the average cycle time for each application along with the number of applications in that month. What was interesting to see was that the number of applications processed decreased from April to June, but the processing time was not affected. COVID may play a part in explaining why the number of applications decreased, but it does not appear to impact the bottom line. This may be accredited to the shift to more of a digital process as mentioned previously.

Next, we looked at exploratory statistics for each carrier. Initial analysis showed that Company A seemed to always complete their applications 10-20 days quicker than the other carriers. Company C on the other hand was 20 days slower on average. Looking at standard deviation, all the carriers were relatively similar with deviation between 41 and 49. Furthermore, we found 761 outliers on the right tail of the distribution in the original data. This showed how much variability we have in our dataset and the need to conduct extensive preprocessing to produce a reliable model. More information can be found in our Jupyter notebook.

Analyzing the distribution of our target variable, we noticed the mean days submitted to final was 73.8 days with a standard deviation of 45.9. Our target variable is heavily skewed to the right which may require a transformation along with removing outliers to make our data more normally distributed. See Figure 1 for the distribution of days submitted to final.

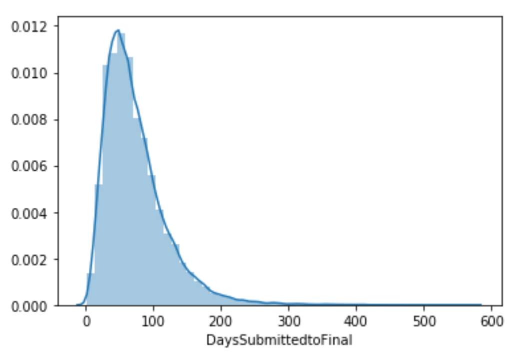

Figure 1

Preprocessing

The dataset did have minor issues that are common in the real world. For example, we saw that the "Insured Age" and "Processing Type" columns contained significant number of null values that had to be addressed. During our discussions in the early stages of the model building process, we had to decide how to handle these null values, and to check whether there are other columns that are not relevant to build our predictive model.

Our intuition told us that the client's age in any life insurance application should be an important factor, and our conversations with our client reaffirmed our hypothesis. Therefore, we concluded that dropping the age as a feature would not be a wise move, so we decided to impute values to replace the nulls based on the median age.

However, we decided to drop "Processing Type" all together, mainly due to the significant null values that were present in the column. Our client explained that "Processing Type" was a relatively new piece of data that was requested from the carriers. Only two of the carriers just recently adopted the "Processing Type" column when providing data to Edward Jones. This resulted in 26,553 null values present within the column even after we filtered for "Final - Inforce" stage only, so we decided to drop the entire column because it was not providing any valuable information at the moment.

For our model that predicts the Days Submitted to Final, we proceeded to drop several other columns from the dataset while building our model, such as "Days In Underwriting", "Days In Approval", "Days In Issued" and "Days In Submitted". Our team members attempted to improve on the prediction model by testing different perspectives towards data cleaning and feature selection, and when these "Days" columns were included during the model training phase, it led to over 90% accuracy. The model's high accuracy was approached with suspicion that led us to find that the model was "cheating" because of data leakage.

The suspect model was provided the multiple "Days" columns that it "has no right to see" (Fraser, 2016), to find the exact "Days Submitted to Final" target value. This is one of the most common mistakes in data science as mentioned by Fraser, and it is called the "No Time Machine Condition," termed by Claudia Perlich who is another data science veteran. Essentially, our model should not be able to travel forward in time to see the actual cycle times after they have occurred and use that data to easily calculate the final outcome. "A model that determines whether it rains at 9:00 AM as a function of whether the road is wet at 9:05 AM might be very accurate, but it will also be very useless" (Fraser, 2016). Since these "Days" columns contained actual processing day values, when added together it would equal our target of "Days Submitted to Final". Therefore, if we included these "Days" columns as features, it would produce misleading results that show during testing that our model is very promising at predicting actual processing time, but in reality it fails to provide any accurate target values.

To further explore our machine learning model, we wanted to build separate models to predict the other steps in the life insurance application approval process. This included "Days in Underwriting", "Days in Approval", "Days in Issued", and "Days in Submitted." In our model we called this a feature named target that we can change and re-run our model. To alleviate the "No Time Machine Condition" with these separate models, we needed to drop all other "days" features besides our target variable.

To determine cycle time SLAs from the dataset that our client provided, we called upon the analytical skills that we learned throughout the MSBA program and in our personal and professional projects. Before we chose a particular method for our predictive models, we began with exploratory data analysis and data cleaning on the dataset provided by our client. Recall that we filtered our dataset to only look at rows with "Final - Inforce" on the "Stage" column, which gives us roughly fifty-one thousand rows of data. We then proceeded to perform basic analysis on the provided CSV data using Microsoft Excel, with more emphasis placed on the target variable “Days Submitted to Final”. We noticed that when “Days Submitted to Final” was sorted in descending order, the “Policy #” column also appeared to sort in descending order. Based on our understanding of “Policy #”, it is an unique value that is considered the row ID of the dataset, which means there should not be any correlation between “Days Submitted to Final” and “Policy #”. We wanted to confirm this assumption, and also perform more advanced analysis, so we shifted our focus towards coding our machine learning model in a Jupyter Notebook using various Python libraries such as NumPy, Pandas, and scikit-learn.

Handling Categorical Values

To be able to utilize categorical features in our model, we needed a way to convert them to numeric features. To do this, we used Pandas’ get_dummies() function to automatically generate dummy values from the columns containing categorical values. For example, the values of “Express Underwriting” and “Full Underwriting” in the “U/W Type” column were converted to a numeric 0 or 1 to denote whether it is full or express underwriting. We also grouped the “U/W Type” of “Reduced Underwriting” with “Full Underwriting” based on a recommendation from our client to align with their reporting.

Handling Null Values

After trimming down our dataset to exclude the outliers, we checked to see what column have high amounts of null values. We used Pandas’ built-in isna() function to count the number of nulls present in each column and we saw 36 columns that contained more than one null value. Many of the columns contained thousands of null values, which made imputing values impractical, so we decided to drop any columns that contained at least 10% null values. For the InsuredAge column, we had a relatively low amount of null values at roughly 2%, so we decided to impute the missing values based on the median of the InsuredAge column.

Log Transform

With a dataset that is clean of null values and outliers, we now wanted to make sure the target variable and features that we will use for the predictive model is relatively normally distributed. Our first attempt to transform the data was using sklearn’s preprocessing.scale() function. This function transforms the data to be more normally distributed, which would help our model’s performance. However, our initial models using preprocessing.scale() produced results that we were not satisfied with, so we performed research on the most popular notebooks on Kaggle to learn how other industry veterans approached data transformation. One method we found was NumPy’s log1p() function, which is calculating the value of log(1+x). The reason we take the log of 1 + the data is because log of 0 is undefined. If 1 is added to all data points, that means the minimum of our data would be 1, and log of 1 is 0. This log transformation helped our data distribution become more normally distributed, and we were able to improve our predictive model.

Correlation

After all the columns in our dataset no longer had null values, we then moved on to check the correlation of the columns compared to the target variable. We used Pandas’ built in corrwith() function to help us visually look at the correlation numbers, and we decided to set a threshold of 1% correlation to keep the column. The reason we chose such a low threshold was to remove some features that had no predictive power in our model, but to keep as many features as possible. This mainly removed some of the new COVID information from our dataset that was irrelevant. When we attempted to set a threshold of features above 10%, 20%, and 30% correlation to the target feature, we noticed a drop in the performance of our model. Since we had numerous categorical features used in our predictive model, we needed as many features as we could use to explain the variability in our target variable.

Removing Outliers

From our exploratory analysis we saw that the mean for days submitted to final was 73.8 and standard deviation was 45.9. Even with the high standard deviation, there were hundreds of outliers in the full dataset. Using the empirical rule of statistics, 99.7% of normally distributed data should fall within three standard deviations from the mean. Rather than eliminating outliers from the original dataset, we used this rule to identify outliers in the split dataframes for each insurance carrier and remove them from their respective models. Since, we had previously normalized some features using log transform, we were able to properly eliminate outliers from both tails of the distributions.

Multicollinearity

Once we had our features selected to use to predict our target variable, we needed to check for multicollinearity. Multicollinearity happens when two independent variables have a high correlation with one another. By having them both in the model, we create redundant information which can skew our overall model performance. To handle this, we looped through each feature in our dataset and checked to see if it had above an 80% correlation with another independent feature. If it did, we saved those values as a key-value pair in a Python dictionary. Once we had a dictionary created for all these multicollinear features, we needed a way to remove them from our dataset. To do this, we checked for the correlation between each of those features that were correlated with one another against the target variable. Whichever feature had the highest correlation with the target we kept and the rest we dropped. This gave us a list of features to drop and remove from our dataset.

Model Building

Now that we had our data cleaned, we were ready to start building a model. As mentioned previously, we had two different models we attempted. One was built to predict regardless of the carrier and the other to predict each carrier separately. For the overall model we just utilized a simple linear regression to predict the target variable. We attempted multiple models and linear regression returned the best average test score. To remove data leakage, we continue to tweak our model using cross validation.

We were unsure which model would perform best for each individual carrier model, so we tested Linear Regression, Decision Tree Regression, XGBoost, and Random Forest Regression. These models were chosen due to their varying methodology and quick processing times (see references for more information). We implemented a Python for loop that allowed our model to pick the best performing machine learning algorithm for each carrier based on the average test score from our cross validation. Once we had our predictions from each carrier, we looked to combine these to predict overall SLA times. Since our R^2 values were relatively low for each model, we needed a different way to predict our SLA times. This required us to take the average results from our predictions and multiply them by a weight. The weight was calculated by taking the percentage of policies in our dataset for that carrier divided by the entire length of the dataframe. We then took the sum product of these weights and our average prediction for each carrier to get a rough SLA for the overall process and to analyze the effect of different features on cycle time.

Placement Rate Analysis

The additional data provided by our client on submitted applications was utilized to analyze Placement Rate metrics. The Placement Rate (PR) is measured as a percentage of applications within a designated timeframe that reach a final status of inforce. Our team was tasked with determining an appropriate SLA for three variations of PR calculations. As a last-minute request, we were also tasked with building a formula that could capture PR accurately as the firm has yet to establish a way to capture this data with the provided data source.

Our approach involved connecting Excel data to Tableau to build visualizations that depict the PR as an aggregate as well as by individual insurance carrier. Tableau was our platform of choice as our client uses this tool primarily with its internal and external reporting. In Tableau, we built calculated fields to determine the stage of each application over a designated timeframe. The percentage of applications in a “Final - Inforce” stage would represent the PR. Additionally, we isolated the population of applications submitted prior to the COVID-19 pandemic as well as the ones submitted after to note any shifts in performance that could be related to the changing working environment. The data available only allowed for a comparison of the three-month PR from September of 2019 through June of 2020. Similarly, the six-month PR only allowed for a comparison of submissions from September of 2019 through March of 2020. The narrowed down population made it difficult to fully assess the “post” COVID environment, but we had enough data to see the overall trend.

Results and Conclusions

Cycle Time-Significance

Looking at the table below, we can see the correlation between each feature and days submitted to final in order from highest absolute value of correlation to lowest. What we found interesting was as COVID cases increased, our days submitted to final decreased. This could be an effect of the efforts taken to expedite the application process in the remote working environment, such as adding electronic capabilities.

Overall Dataframe			
Feature	Correlation	Feature	Correlation
#ofAPS?	0.387855	RiskClass_PreferredPlus	-0.130706
Exam_Yes	0.364884	PlanType_Term	-0.108133
U/WCombined	0.323155	RiskClass_Rated/SubstandardRisk	0.107702
new_cases_smoothed	-0.264824	RiskClass_PreferredNon-Smoker	-0.081792
1035?_Yes	0.250686	Interview_Yes	-0.056899
PlanTypeGroup_Perm	0.221523	PlanType_UL	0.03672
PlanType_Hybrid	-0.200632	RiskClass_StandardSmoker	0.035825
stringency_index	-0.200231	PlanType_SVUL	0.03564
PlanType_VUL	0.176651	RiskClass_PreferredSmoker	0.0274
RiskClass_StandardNon-Smoker	0.149478	PlanType_SUL	0.020279
InsuredAge	0.14919	Benefit	0.01079
new_deaths_smoothed	-0.132601	RiskClass_StandardPlus	0.010556

Figure 2

The table of correlation to “DaysSubmittedtoFinal” also confirms that the SLAs Edward Jones has in place currently are based on appropriate metrics. As seen in figure 2 above, features related to APS, exams, underwriting type and 1035 exchanges had the most significant impact on the overall cycle time, so the attention given to each of these variables is fitting for the SLAs addressed in the next section of our report.

Another detail that caught our attention was that the rise of COVID-19 cases had a correlation with a decrease in cycle time as noted in Figure 2. Considering recent events, the shift to a remote-working environment off-and-on in 2020 has forced Edward Jones and other businesses to adjust with new technologies and capabilities to serve customers. We found that over multiple iterations, the use of the full original dataset produced a weaker model. It is likely that these changes in processing led to quicker cycle times which contributed to the lower predictability of our model when the full dataset was included. As a result, we addressed this variance by excluding the data prior to COVID-19. This also ensured a similar product mix as our client made us aware of a significant change that occurred in December of 2019 that may skew the data further.

Cycle Time

The predictive model homed in on submitted applications in 2020, narrowing down the population to a more consistent product mix based on products more likely to be offered in Edward Jones' current selection. Outliers were removed from the model and specific metrics such as APS had additional filters applied to align with the current format of Edward Jones SLAs. For example, the cycle time "With 1035 Exchange" was measured from the population of non-term cases without an APS.

Figure 3 below compares the cycle time values predicted by our model versus the actual result on the test set. Our model was successful in predicting the cycle time impacts of highly correlated features, such as APS, exams, underwriting type and 1035 exchanges. However, the model was not successful in accurately predicting the cycle time by stage or with the less correlated features such as interview. The variability of each insurance carrier's cycle times prevented our model from performing at a high enough level to establish SLAs with predicted values. As a result, our SLA recommendations based on our analysis are in the "Actual" column in Figure 3 below.

	Cycle Time	
	Prediction	Actual
Days in Submitted	0.9	1.11
Days in Underwriting	31.24	36.8
Days in Approval	4.71	8.09
Days in Issued	18.1	12.47
Days Submitted to Final:		
Full Underwriting	53.43	58.97
Express Underwriting	33.89	38.27
Change in Cycle Time With:		
APS	+19.35	+20.21
Per APS	+14.71	+15.37
1035 Exchange	+25.82	+27.79
Exam	+4.93	+5.33
Interview	+2.47	+5.58

Figure 3

After running five models with the target set to the days in each stage, we found that the reason for the lack of predictability fell on two issues:

1. There were not enough data points to explain the unique aspects of the insurance application process that lead to the overall cycle time. Specifically, the models predicting days in submitted, days in approval and days in issued performed poorly due to the lack of data correlated with the target variable.
2. Multicollinearity between various features reduced the number of available data points to predict the target variable.

It is worth noting that the use of the "actual" values is different from simply taking the average of the unfiltered data. Our model provided insights into a dataset that has run through pre-processing and was weighted based on each carrier's individual model and cases. Additionally, our model was joined with COVID-19 case related data, narrowing the scope to 2020 data.

Placement Rate

Our team came up with a solution to calculate the three-month PR and six-month PRs using Tableau. We took a more visual approach to address the SLA as the predictive model we built for cycle times was not applicable to this metric. As an added benefit, this calculation can be integrated with existing Edward Jones reporting to eliminate two data sources in use today by leveraging existing data to streamline the process.

Using the available ten-month period, the PR metrics were compared month-over-month and segmented before and after the start of the COVID-19 pandemic. As indicated by Figure 4, there has been an increasing trend since March of 2020 indicating a 41.3% average Post-COVID.

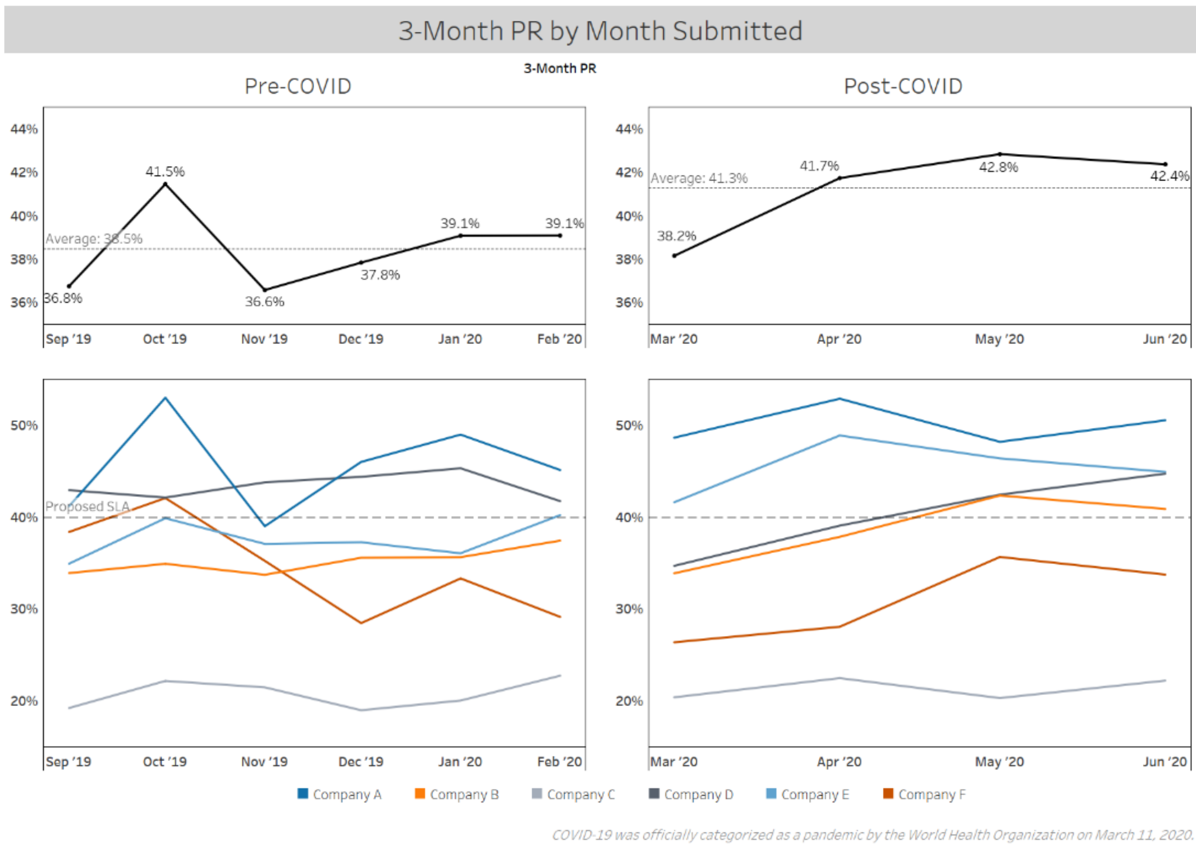

Figure 4

The same logic was applicable to the six-month PR. Using the available seven-month period, the PR metrics were compared month-over-month and segmented before and after the start of the COVID-19 pandemic. As indicated by Figure 5, February and March of 2020 achieved a 57% result. Based on the three-month placement trend and assuming a similar product mix has existed in the latter months, it is likely that the results will be higher in the months following March once the data is available.

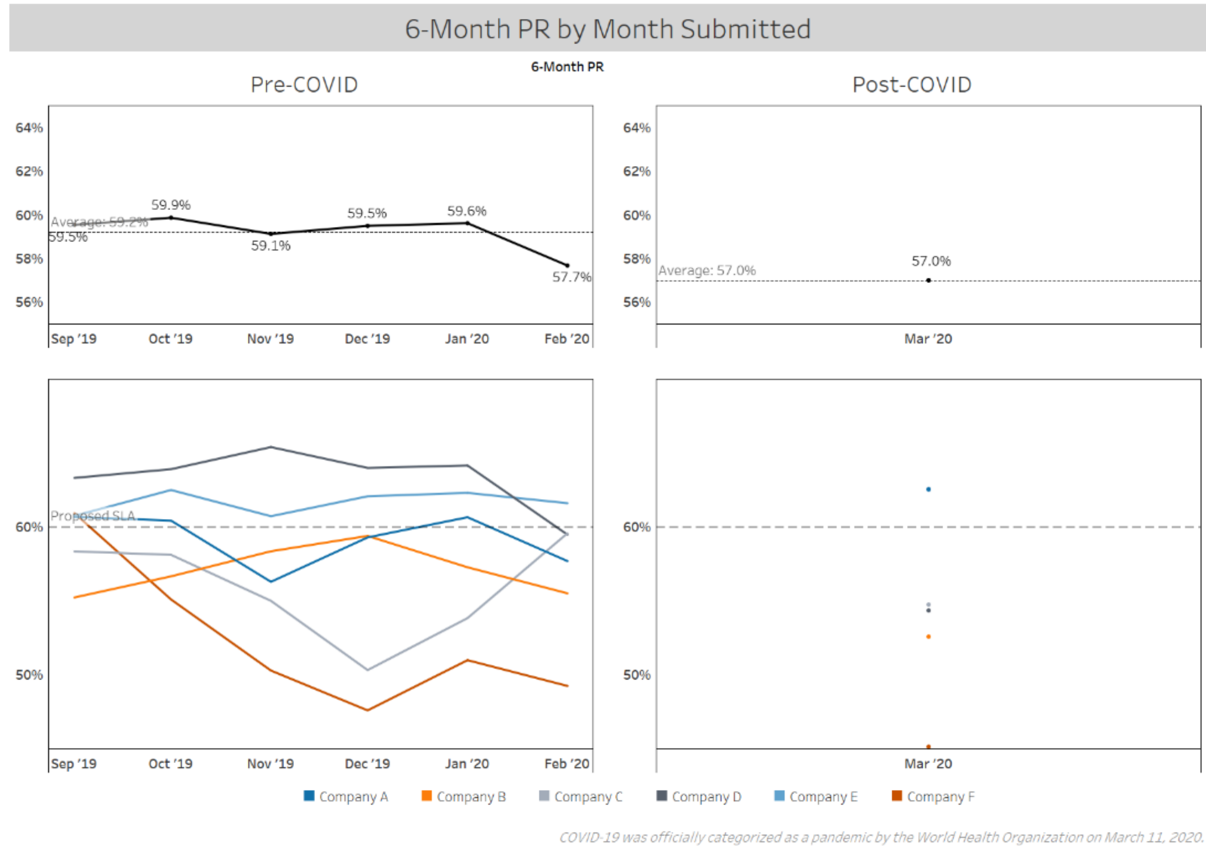

Figure 5

The proposed SLAs are included in figures 4 and 5 based on the recent trends in data and spread by insurance carrier. A 40% SLA is proposed for the three-month PR, and a 60% for the six-month PR. However, our team found the logic used to calculate PR misleading. Currently, the three-month PR is tracking the progress of individual cases after 60 to 90 days. The six-month PR measures the progress of cases after 150 to 180 days. To accurately assess the cases progress after three or six months, a different calculation would be needed. Our team came up with a calculation that identifies the stage of an application after exactly three months or six months, which could be used as a replacement for the current PR metrics. If Edward Jones adjusts to the new logic, the SLAs would be 50% and 60% respectively (see figure 4 and 5). A PDF breakout of the revised PR metrics will be provided for further considerations if desired.

The final point of analysis regarding PR was to determine the SLA for an "overall" PR. Simply looking at the final-inforce population compared to the final-not active population, there is approximately a 67%/33% split. The parameters around an "overall" PR would need to be clearly defined by the firm, but a starting point for an SLA could be 67% based on the known population of recent data. The challenge with this request is the variance within the data. The one-year population of cases is a good starting point, but it may be worth drilling down to subsets of the data to determine the impacts. For example, the hybrid population had a 63% overall PR, whereas the perm population was at 70%. Rather than indicating how many cases are placed over a specific amount of time as with the three-month and six-month PRs, the overall PR provides a baseline for the general likelihood of achieving placement, given that a final outcome has occurred.

Future Efforts

For future efforts in improving our model, we should address some of the noise found in our model along with adding more data provided by Edward Jones. See below for further explanation.

Eliminating Noise

There were multiple features that we needed to include in our model that had multicollinearity with one another, but we needed to keep these features in the model to be able to address some of the questions posed by Edward Jones. In our model, these features were saved to a Python list named "kept_feats." The features included in this list were "PlanType_Term", "PlanTypeGroup_Perm", "Exam_Yes", "U/WCombined", and "PlanType_VUL." These features were being removed during the preprocessing phase of our model, but we called them to return SLA times for our model when an exam was present or not, etc. By allowing our model to remove these features, we may improve our overall model.

Adding Data

We believe additional features could be added to our dataset to improve our model's performance in predicting the length of the cycle time of an application. Depending on the level of data availability, these include:

- Delivery/Submission of application
- Number of 1035's per policy/conservation periods
- What vendors conducted the exams/APS/interviews
- Policy premium information
- Associated applications

For the delivery/submission of an application, if an application were submitted via paper, we would expect delays in our cycle time due to delivery. Whereas if an application is digital, we would not need to wait for this delivery.

When it comes to 1035's, there are multiple items that would be good to know. The first element would be understanding the number of 1035's per case. If there are multiple 1035's, we expect the process to take longer due to multiple companies having different processes. It would also be helpful to know the ceding companies that are processing the 1035's and their respective conservation periods. By understanding the expected turnaround time on a 1035, we can better understand the time it takes to complete the 1035.

For requirements collected by vendors such as exams, APS, and interviews, it would be helpful to know the vendor collecting this information. Certain vendors may have different processing times that prolong the application processing time.

Policy premium information would also be helpful to know. If we had the premium for a policy, it may assist us with understanding how the cost of a policy effects processing time. The more expensive a policy, we assume the lengthier the process.

The last feature that would be helpful to gather would be whether a case has an associated application. If a couple has two separate life insurance applications, they would be looking to be approved through each stage concurrently. If one of them needs to have extensive paperwork completed while the other is already approved, this may help explain why some cases are taking much longer than expected.

References

Developers, S.-L. (2007). *Random Forest Regressor*. Retrieved from Sci-kit Learn : https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Developers, S.-L. (2007). *Sci-kit Learn*. Retrieved from Decision Tree Regressor: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

Developers, X. (2020). *XGBoost Documentation*. Retrieved from ReadTheDocs: https://xgboost.readthedocs.io/en/latest/

Fraser, C. (2016, August 24). *The Treachery of Leakage*. Retrieved from Medium: https://medium.com/@colin.fraser/the-treachery-of-leakage-56a2d7c4e931

Horbal, M. (n.d.). *The Only Paramedical Exam Resource You Will Ever Need*. Retrieved from RiskQuoter: https://www.riskquoter.com/underwriting/paramedical-exam/

Kagan, J. (2020, June 23). *Section 1035 Exchange*. Retrieved from Investopedia: https://www.investopedia.com/terms/s/sec1035ex.asp

Papiu, A. (2016). *Regularized Linear Models*. Retrieved from Kaggle: https://www.kaggle.com/apapiu/regularized-linear-models

Ritchie, H. (2020). *Coronavirus Source Data*. Retrieved from Our World in Data: https://ourworldindata.org/coronavirus-source-data

Appendix – Reproduction of Results

- We submitted a compressed zip file on our Canvas submission. Please download **MSBACapstone-main.zip** file to your PC, right click on the zip file and click Extract All…
- Location of Data Source –within the zip file: **ASU_Team10Project_InsuranceDataset.csv**
    - GitHub for all our dataset and code is also available at: https://github.com/jluczak18/MSBACapstone
- Tools/Platform used to run analysis: Python, Pandas, SKLearn, NumPy, MatPlotLib, Tableau, Excel, Anaconda, Jupyter
- List of scripts used to run analysis: located within the zip file: **MSBA_JupyterNotebook.ipynb**
    - For model output on different application stages, please see the following Jupyter Notebooks:
        - **DaysinApprovalModel.ipynb**
        - **DaysinIssuedModel.ipynb**
        - **DaysinSubmittedModel.ipynb**
        - **DaysinUnderwritingModel.ipynb**
- Guidance on how to run the scripts: please follow guide located within the zip file: **Model Implementation Guide.docx**
- For our placement rate analysis, please see the Tableau file **ASU_Team10_AppliedProject.twbx** or **NewPlacementRateMetrics.pdf**

Metadata

COVID-19 Dataset

New cases smoothed – 7 day rolling average of new COVID-19 cases

New deaths smoothed – 7 day rolling average of new deaths

Stringency index - Government Response Stringency Index: composite measure based on 9 response indicators including school closures, workplace closures, and travel bans, rescaled to a value from 0 to 100 (100 = strictest response)

ASU_Team10Project_InsuranceDataset.csv
Carrier – the client's life insurance partners
Policy # - the unique ID of the life insurance application
U/W Type – the type of underwriting performed for the application
Plan Type – the type of life insurance product for the application
Plan Type Group – the category of the life insurance product for the application
Term Duration (Yrs) – for a term plan type only, the duration of the policy
Benefit – the amount of payout for the life insurance policy
Stage – the current status of the application process
Submitted Date – the date the application was submitted
Days in Submitted – the number of days the application required from submitted to the start of the underwriting process
Underwriting Date – the date the application began the underwriting process
Days in Underwriting – the number of days the application required from starting the underwriting process to application approval
Approval Date – the date the application gained underwriting approval
Days in Approval – the number of days the application stayed in approval until it was issued
Issued Date – the date the application completed all delivery requirements, such as collecting signatures and paying the policy premiums
Days in Issued – the number of days the application stayed in issued until it became inforce
Inforce Date – the date application completed all processes, and the premiums are paid
Final - Not Active Date – the date the application was closed due to various factors
Days Submitted to Final – the total cycle time of a successful application process from beginning to end
Processing Type – an indicator of the type of application (I.e. conversion, reissue, new application)
Insured Age – the age of the proposed insured at the time of submission
Risk Class – the level of risk assessed to the insured as of the last status update
Product – the specific name of the product
# of APS? – the number of APS ordered throughout the application process on an inforce policy
1035? – an indicator of whether an inforce policy included a 1035 exchange
Interview – an indicator of whether an inforce policy required an interview on the insured
Exam – an indicator of whether an inforce policy required an exam on the insured
Requirement Group 1 – calculated field that checks for 1035, APS, both, or neither
Requirement Group 2 – calculated filed that checks for interview, exam, both, or neither