Gayathri Gopalan
Fractal.ai

Approach:

Problem Statement:

Smart Agent Recruitment for FinMan Distribution Company

Understanding the Problem Statement- Deriving meaningful Insights using PowerBI Dashboards:

Initially I used PowerBI to gather insights and visualize the data by creating a Dashboard. The overall recruitment process is analyzed Applicant-wise and Manager-wise. Applicant age, Manager Age was grouped into bins to comprehend the relationship between manager and applicant. There were a higher number of Middle-Aged managers whose employment status was confirmed and contributed to the application registration. Out of 75% of male applicants from the total, 66% were married and contributed to business sourced

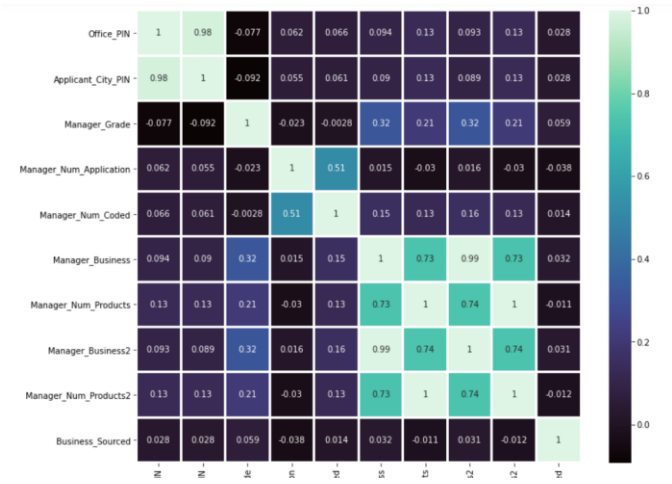
Business Sourced by applicant within 3 months of recruitment, the target variable, is the factor that contributes to identifying the right agents to hire.

Predicting the Target Variable for each Potential Agent- Model Building:

(a) Exploratory Data Analysis (EDA):

The initial step is **Hypothesis Generation**. Here we question which agents/applicants are more likely to bring in more business sourced.

This is followed by the **Descriptive and Inferential Statistics** where the nature of independent features of the data are investigated- datatypes (categorical, numerical and datetime). From the visualizations (KDE plots), mean and median, we can find that most numerical variables are right-skewed and have a very less (<6%) correlation with the target variable. This will make the prediction less reliable, so it would be necessary for us to create highly correlated features.



Conducting Univariate and Bivariate Analysis and visualizing the categorical variables.

I used mode() to fill the missing numeric values, and imputed missing values of categorical variables with “other”/”unknown” and performed Label Encoding. For the date type columns, I imputed the missing data values with ‘1/1/1900’ (a random date value), created a column called ‘today’ and calculated the current date.

Initially, I did create a difference of date columns for all by subtracting it from current date (‘today’) and converted it into numeric datatype in order to include date values/pattern in the model. For eg, if DOJ= 2/2/2007, the date_diff value would be 12/7/2021- 2/2/2007

(b) Feature Engineering:

In order to figure out which variables would be well correlated with the target variable; the following variables were created:

- **ApplicantAge**: Age of the applicant (when application was received)
- **ManagerExperience**: Experience (duration) of the Manager (when application was received)
- **ManagerAge**: Age of the Manager (when application was received)
- **Manager_Business2_categoryA** and **Manager_Num_Products2_categoryA**: Segregating Category A advisor from amount of business sourced and no. of products sold from the 2nd column
- **Agent_proximity:** Proximity of agent's location from work
- **Application_CVR:** Application conversion rate of manager - No. of recruitments
- **Manager_Level_Hops**: Manager current designation- Manager joining designation (after converting them to numeric and removing the strings)
- **Manager_Growth**: Binning Manager Level Hops into High, Medium and low for visualization
- **Applicant_rank**: Rank the order in which applications were received in a single day

(c) Data Preprocessing

- *Data normalization on Date columns:* Data normalization transforms numeric columns to the same scale and improves the learning algorithm/performance of the model when there are unordered data. So I did perform min() max() scaling on date columns.
- *Standardizing the dataset:* Using StandardScalar() and fit_transform() to standardize the data
- *Imblean:* After visualizing the Target variable, I used the imblean smote technique to balance the distribution

*(d) **Model Building***

After splitting the train-test sets from train (80-20) with a Random state of 30 and stratify as 'y', I started with a Logistic Regression without excluding any feature (other than applicant_rank), fetching the accuracy of the model and plotting the ROC curve. The accuracy of the model obtained was 58% which was just on par with the expected accuracy.

Then I removed the columns which were having a correlation of less than 1% namely - 'Application_Receipt_Date', 'Applicant_BirthDate','Manager_DOJ', 'Manager_DoB', 'Manager_Business2_categoryA', 'Manager_Num_Products2_categoryA', 'Agent_proximity', 'Manager_Level_Hops', 'today'. Using imblean did not contribute with improving the model, rather introduced more loss - therefore, removed the SMOTE approach. After all these changes I was able to achieve 61% accuracy. Then I tried using various models like XGBM, Random Forest Tree, LGBM but I could not improve the accuracy. Suspecting overfitting, I introduced Stratified K-Fold cross validation approach, but it did not help much. Since all the features are related to the manager's performance, but there are not many variables to explain applicant's nature/performance – I tried to create some features around applicant info. Multiple applications were received in a day, my winning feature was calculated by ranking the applications within a day. This was having ~50% correlation with the target variable. It boosted my accuracy from 60% -> 79% using the same logistic regression model.

(e) Future Implementation

More historical data with less missing values would have better impact in the performance of the data. Having a column or some way to uniquely identify manager may help the analysis, by mapping it with the applicant information.

By removing some more less correlated fields and introducing more features around applicant performance can further boost the data.