SUPERVISED LEARNING: CLASSIFICATION

Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation and the benefits that your analysis provides to the business or stakeholders of this data.

Main objective with this work is to model for prediction purposes.

Benefits:

- Main aim for this project from a business perspective is to reduce cost the company.
- Context here is that a company is hiring people for sales and training them so a huge upfront cost is involved.
- This model tries to predict during the time of application itself whether an applicant will turn out to be a good sales agent or not.

Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

The data I took is about hiring new associates/ salespeople by managers in sales division. Since the company is making upfront investment in hiring and training, it tries to predict based on past data if an applicant will produce business after getting hired in future.

So, the data is made up of 14572 rows (each of a distinct applicant) and 22 features for prediction and one binary target variable of which training data has 9527 entries, and test has rest. So we have 14572 people who applied in the past and got hired and whether they sold a product within three months is reflected in business sourced target column. The features/ variables are:

- Identification: 'ID'
- Location of Offices: 'Office_PIN'
- Applicant Details: 'Application_Receipt_Date', 'Applicant_City_PIN', 'Applicant_Gender', 'Applicant_BirthDate', 'Applicant_Marital_Status', 'Applicant_Occupation', 'Applicant_Qualification'
- Manager details: 'Manager_DOJ', 'Manager_Joining_Designation', 'Manager_Current_Designation', 'Manager_Grade', 'Manager_Status', 'Manager_Gender', 'Manager_DoB', 'Manager_Num_Application', 'Manager_Num_Coded', 'Manager_Business', 'Manager_Num_Products', 'Manager_Business2', 'Manager_Num_Products2'
- Target: 'Business_Sourced'[1]

Here, variable types are:

- Date: 'Application_Receipt_Date', 'Applicant_BirthDate', 'Manager_DOJ' and 'Manager_DoB'
- Numerical: 'Manager_Num_Application', 'Manager_Business', 'Manager_Num_Products', 'Manager_Business2', 'Manager_Num_Products2'

[1] For the rest of the document, successful means Business sourced is 1, otherwise 0.

Categorical: rest all

I am trying to model the data to predict whether an applicant now will become successful in future or not, ie, Business sourced is 1 or not.

Brief summary of data exploration and actions taken for data cleaning and feature engineering.

1. Handling date variables
    a. Since we cannot handle date variables directly, we can get the information out of it by creating new columns.
    b. To see if month or year of application date has any effect in determining target, creating separate columns each for month and year instead of date directly
    c. From applicant's DoB, creating a variable for applicant's age since I believe month and date wont have any meaningful impact on their success of work
    d. Similarly using 'Manager_DOJ' and 'Manager_DoB' to calculate manager's age and experience in the company
2. Handling non-numerical variables:
    a. PIN codes were used as geocodes to identify distance between applicant address and office location.
    b. Gender and Manager Status (whether probation or confirmed) were binary encoded.
    c. One Hot encoding (encoding with indicator variables) was performed for: 'Applicant_Marital_Status', 'Applicant Occupation'
    d. Label encoding (Rank ordering) was performed for: 'Manager_Joining_Designation', 'Manager_Current_Designation', 'Manager_Grade'. Grade was already ordinal; designation was of format "Level xx" was converted to ran based on xx.
    e. Target encoding was performed for: 'Applicant_Qualification'
3. Handling missing values
    a. I noticed something in the data wrt missing values, ie, for 683 observations there was no information on any column describing manager detils. So I figured these entries must mean, they had no manager, So created a new column for indicating the presence of manager.

    b. Since every other variable with missing values except birthdate was a categorical variable and #missi ng is less than half a percent of overall data, observations with birthdate missing were dropped and others excluding Applicant_Occupation were imputed with KNN iteratively with Applicant_Gender h aving least missing imputed first. Due to large number of missing values, Applicant_Occupation missi ng was treated as a new category.

4. Handling outliers
    a. 'Manager_Num_Products' had unproportionally large count of 0 and very low counts of extremely high values. Hence, they were floored and capped at 10, 90 percentiles repectively and indicator for flooring and capping was added as a new feature
    b. 'Manager_Business' also had a couple of negative values and some extremely high values. Hence, they were capped and floored at 95, 5 percentiles respectively.
5. Variable Transformation
    a. 'Manager_Business' was heavily skewed towards left hence a log transformation was applied.
    b. 'Manager_Business2', 'Manager_Num_Products2' are essentially second line of business of a manager if exists. So, when it doesn't the data replicates the first and replicates them which leads to extremely high correlation between them. So instead of these, their difference from 1st line of business, ie, 'Manager_Business', 'Manager_Num_Products' were calculated and used to create new columns and dropping these.

c. A new column for growth of manager was calculated as a difference of 'Manager_Joining_Designation' & 'Manager_Current_Designation'
d. Applicant_Qualification was re-grouped into Class X, Class XII, Graduate, Certified Associate, Masters & Others and target encoded.

Summary of training at least three different classifier models, preferably of different nature in explainability and predictability.

The modelling methods which I too into consideration were: 'Logistics regression', 'Decision tree', 'Support Vector Machine', 'Random Forest', 'K Nearest Neighbours'. Note that all of these we modelled and predicted on same train & test data.

The exact hyper-parameters used for each method were:

1. ***LogisticRegression****(solver='newton-cg',max_iter=500,n_jobs=-1)*
2. ***DecisionTreeClassifier****(max_depth=7, min_samples_split=50, min_samples_leaf=10,min_impurity_decrease=0.0001)*
3. ***SVC****(probability = True,max_iter=500,break_ties=True)*
4. ***RandomForestClassifier****(oob_score=True,max_depth=7, min_samples_split=50,min_samples_leaf=10,min_impurity_decrease=0.0001,n_jobs=-1)*
5. ***KNeighborsClassifier****(n_neighbors=10,n_jobs=-1)*

The results I got were:

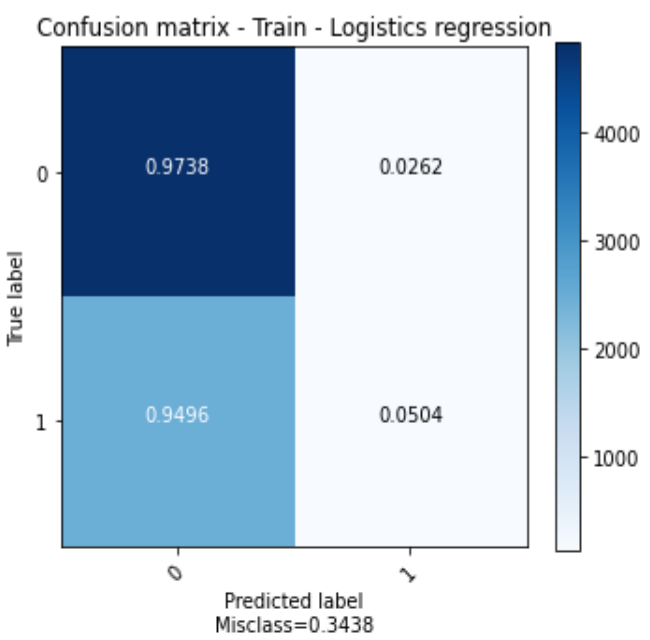
----------Training Data----------

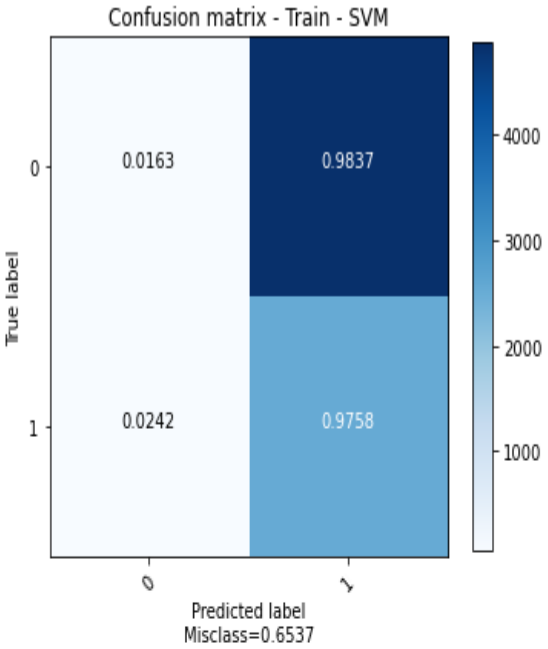
	accuracy	cohen_kappa_score	f1_score	precision_score	recall_score	roc_auc_score
Logistics regression	0.656221	0.030747	0.091544	0.501916	0.050365	0.512083
Decision tree	0.680550	0.157208	0.300116	0.608696	0.199154	0.566022
SVM	0.346291	-0.005482	0.506587	0.342095	0.975779	0.496051
Random Forest	0.666006	0.039328	0.061664	0.912088	0.031911	0.515149
KNN	0.683327	0.158393	0.292467	0.631378	0.190311	0.566034

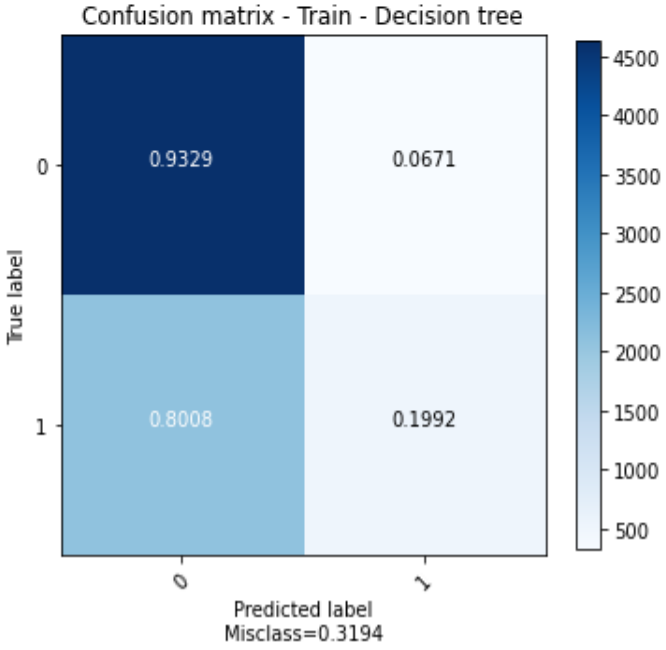
----------Test Data----------

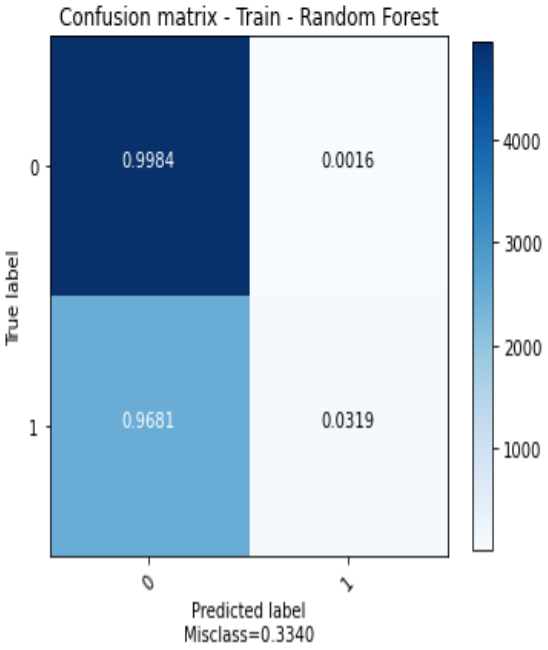
	accuracy	cohen_kappa_score	f1_score	precision_score	recall_score	roc_auc_score
Logistics regression	0.643575	0.017638	0.081744	0.441176	0.045045	0.507012
Decision tree	0.632470	0.043491	0.202067	0.429268	0.132132	0.518311
SVM	0.352195	-0.008273	0.514467	0.349488	0.974474	0.494176
Random Forest	0.645690	0.000207	0.014706	0.357143	0.007508	0.500080
KNN	0.631941	0.029079	0.169451	0.412791	0.106607	0.512079

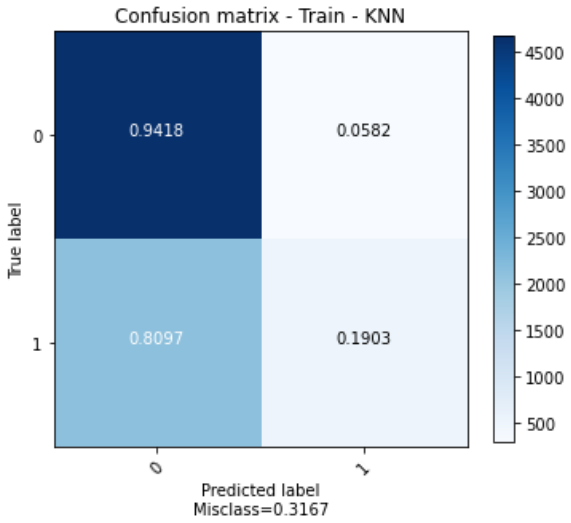
And the confusion matrix for each data sample for each method is shown below:

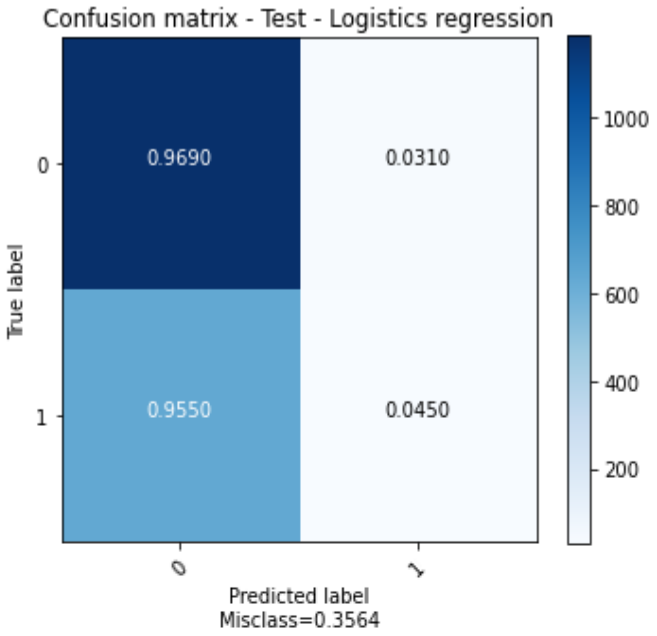

Confusion matrix - Train - Logistics regression
0.9738
0.0262
0.9496
0.0504
True label
Predicted label
Misclass=0.3438

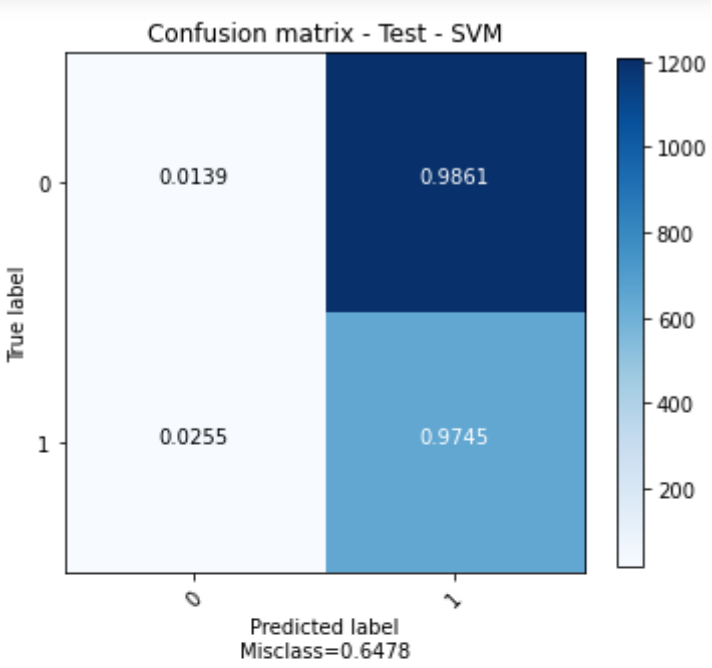

Confusion matrix - Train - SVM
0.0163
0.9837
0.0242
0.9758
True label
Predicted label
Misclass=0.6537

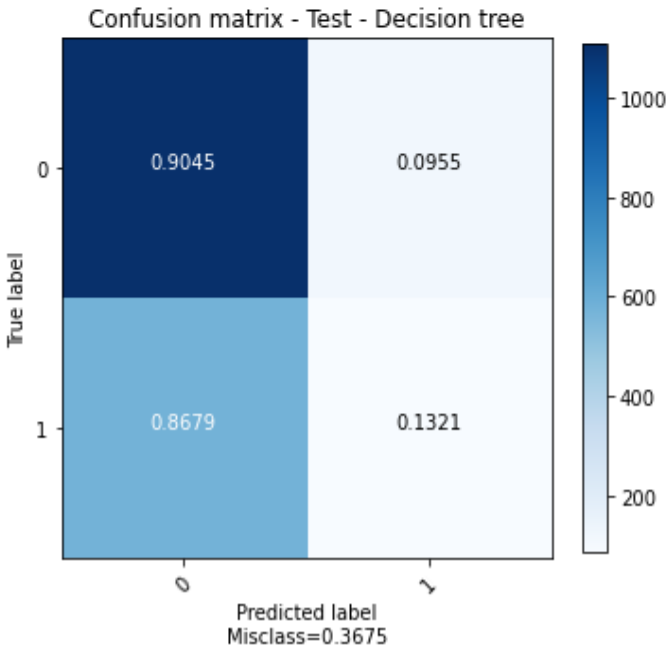

Confusion matrix - Train - Decision tree
0.9329
0.0671
0.8008
0.1992
True label
Predicted label
Misclass=0.3194

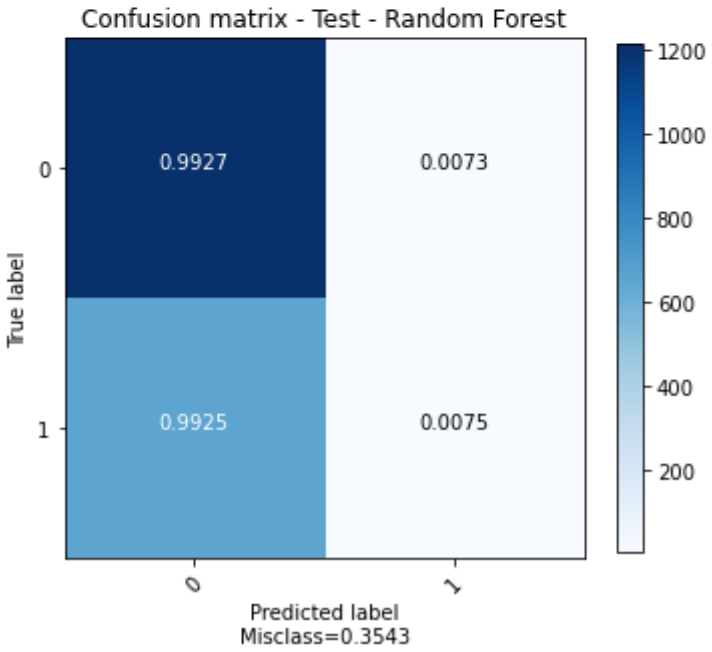

Confusion matrix - Train - Random Forest
0.9984
0.0016
0.9681
0.0319
True label
Predicted label
Misclass=0.3340

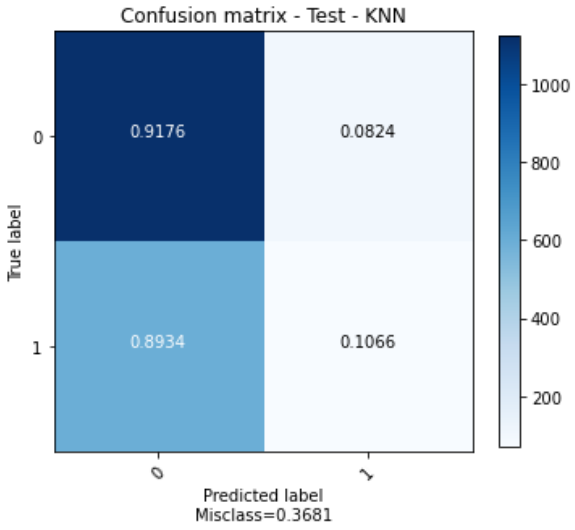

Confusion matrix - Train - KNN
0.9418
0.0582
0.8097
0.1903
True label
Predicted label
Misclass=0.3167


Confusion matrix - Test - Logistics regression
0.9690
0.0310
0.9550
0.0450
True label
Predicted label
Misclass=0.3564


Confusion matrix - Test - SVM
0.0139
0.9861
0.0255
0.9745
True label
Predicted label
Misclass=0.6478


Confusion matrix - Test - Decision tree
0.9045
0.0955
0.8679
0.1321
True label
Predicted label
Misclass=0.3675


Confusion matrix - Test - Random Forest
0.9927
0.0073
0.9925
0.0075
True label
Predicted label
Misclass=0.3543


Confusion matrix - Test - KNN
0.9176
0.0824
0.8934
0.1066
True label
Predicted label
Misclass=0.3681

A paragraph explaining which of your classifier models you recommend as a final model that best fits your needs in terms of accuracy and explainability.

Based on summary statistics of results, we can infer the following:

1. For all the methods, with respect to all comparison metrics, train and test data perform similarly.
2. SVM has uniformly poorer results across scores like accuracy, precision, cohen's kappa than other methods.
3. KNN and tree based methods gives better performance with training data over Logistic Regression.
4. Tree based methods gives better performance with test data over KNN and Logistic regression.

Therefore, now I am comparing between tree based methods.

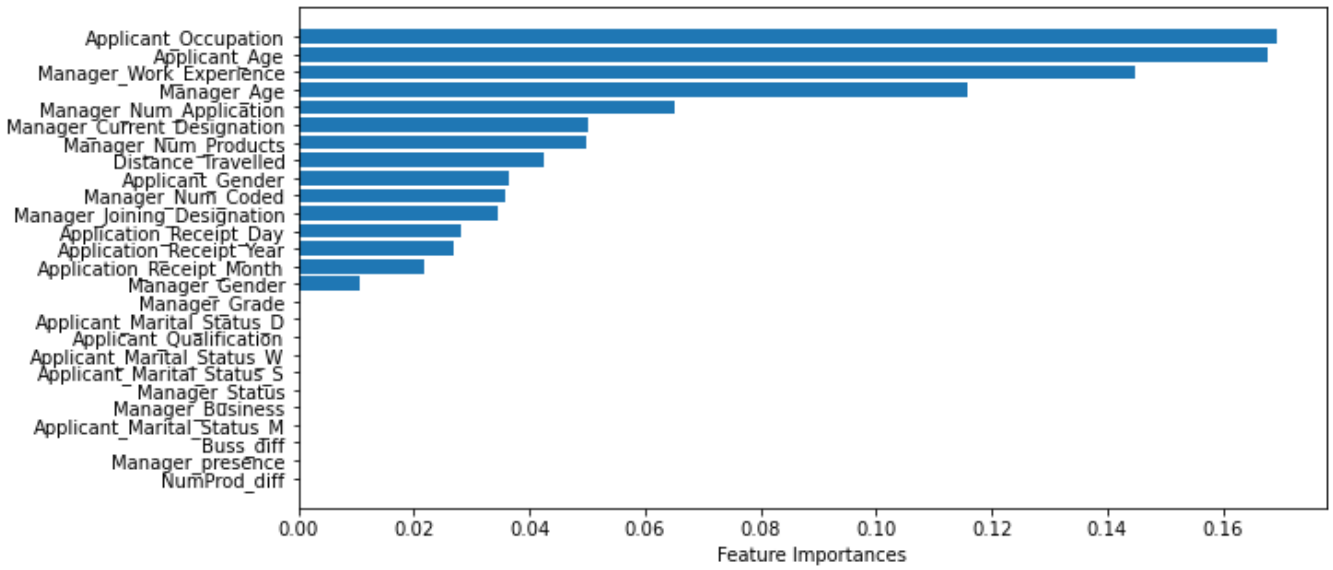
With respect to accuracy and roc-auc metrics, both have similar scores in both train and test whereas in precison, RF has better results than decision tree in train data and Decision tree has better results in both precison and recall in both train and test.

Now, looking at visual ease of understanding and from a perspective of explainability, Decision tree is better than random forest.

Hence, I will recommend to use Decision tree model with parameters as mentioned above.

Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your classifier model.

Feature importances based on the final tree is plotted here:



As we can infer from above plot, an applicants' current occupation and their age are the most important drivers in determining success of an applicant having very close importance values with Manager's work experience, age and number of applications they received coming next.

Suggestions for next steps in analysing this data, which may include suggesting revisiting this model after adding specific data features that may help you achieve a better explanation or a better prediction.

Next step would be to do Hyper parameter tuning and cost complexity pruning and also try a bagged or boosted version of decision tree to improve its overall accuracy and precision.

Additional data regarding performance of applicant in their occupation would give better insights into the overall capacity of the applicant. Also, data on whether that applicant has previous experience in sales would also be a good predictor I believe.