UNSUPERVISED LEARNING

Main objective of the analysis that also specifies whether your model will be focused on clustering or dimensionality reduction and the benefits that your analysis brings to the business or stakeholders of this data.

Main objective with this work is to model for clustering 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 group applicants during the time of application itself whether an applicant will turn out to be a good sales agent or not based on application features.

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 cluster 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 future - looking binary variable[1] 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'
- Performance comparison Variable: 'Business_Sourced'

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] This variable is just used in comparison metric and not during clustering

Categorical: rest all

Outline: The company is having some sales agents now and their applications so they try to cluster them and score them based on their current performance to check which clustering model works best for them so that they can group applicants in future.

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 won't 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 #missing 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 having least missing imputed first. Due to large number of missing values, Applicant_Occupation missing 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.
e. Finally, both train and test data were first standard scaled to preserve distribution and then min-max scaled to bring it to default 0-1 range.

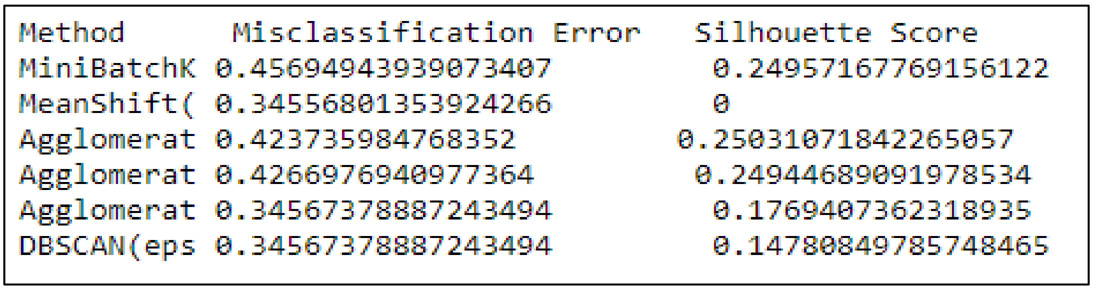
Summary of training at least three variations of the unsupervised model you selected.

The clustering methods which I took into consideration were: KMeans, MeanShift, AgglomerativeClustering with different metrics and linkages, DBSCAN. Note that all of these we modelled and predicted on same train & test data. Hyperparameters and final results were obtained through grid search and 10-fold cross validation.

The exact hyper-parameters used for each method were:

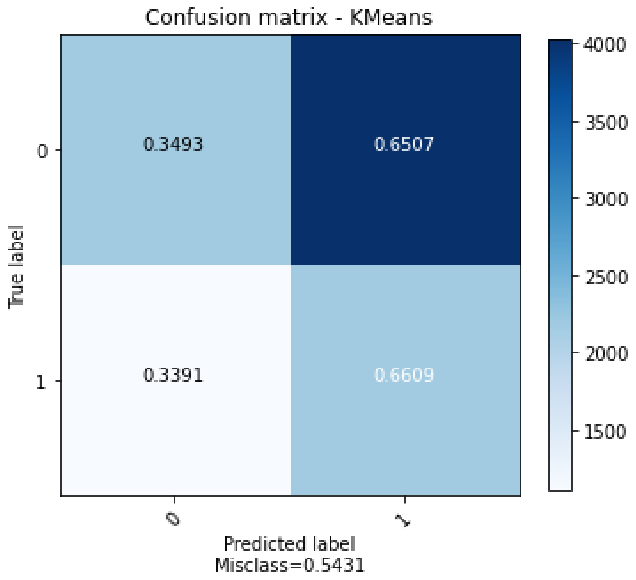
1. ***MiniBatchKMeans****(n_clusters=2, max_iter=500, max_no_improvement=None, init_size=30, n_init=3)*
2. ***MeanShift****(n_jobs=-1, max_iter=500)*
3. ***AgglomerativeClustering****(n_clusters=2, affinity='****euclidean****', linkage='****ward****')*
4. ***AgglomerativeClustering****(n_clusters=2, affinity='****l2****', linkage='****complete****')*
5. ***AgglomerativeClustering****(n_clusters=2, affinity='****l1****', linkage='****single****')*
6. ***DBSCAN****(eps=1.25, n_jobs=-1)*

The results I got were:



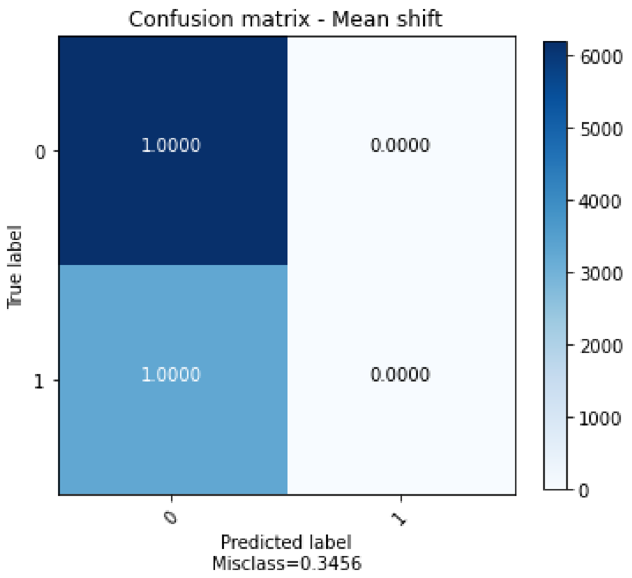
```
Method      Misclassification Error   Silhouette Score
MiniBatchK 0.45694943939073407         0.24957167769156122
MeanShift( 0.34556801353924266         0
Agglomerat 0.423735984768352          0.25031071842265057
Agglomerat 0.42669769409773 64         0.24944689091978534
Agglomerat 0.34567378887243494         0.1769407362318935
DBSCAN(eps 0.34567378887243494         0.14780849785748465
```

And the confusion matrix for each method with actual clusters known is shown below:



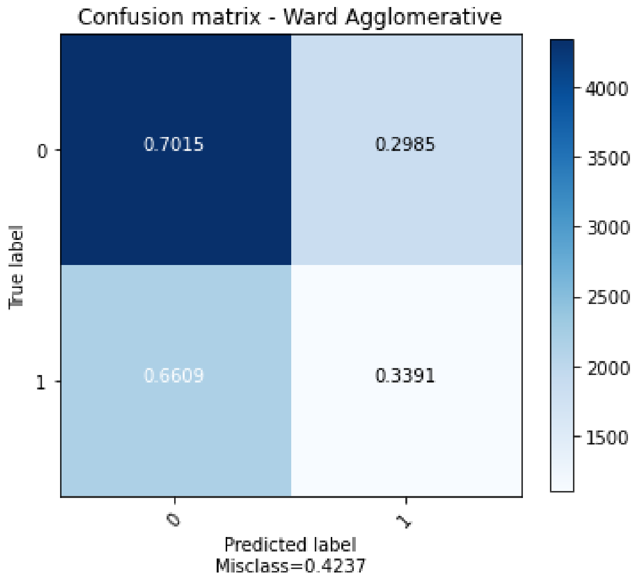
Confusion matrix - KMeans

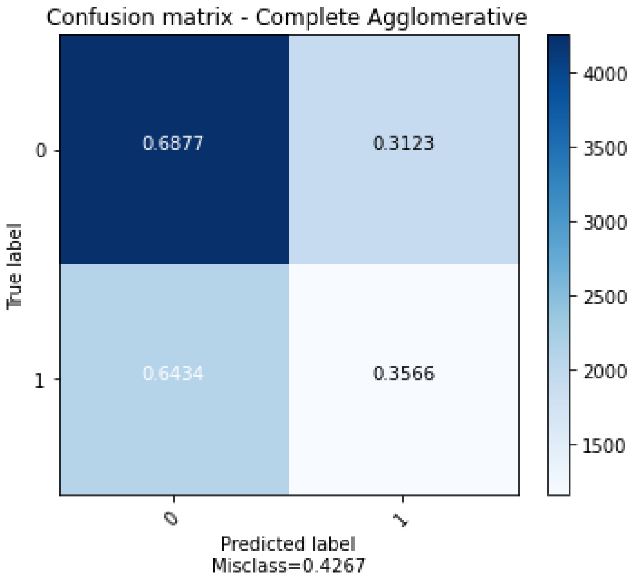
Misclass=0.5431

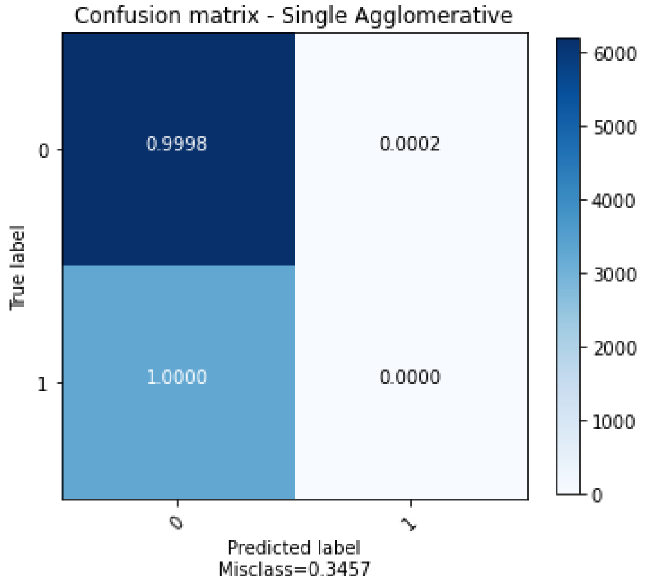


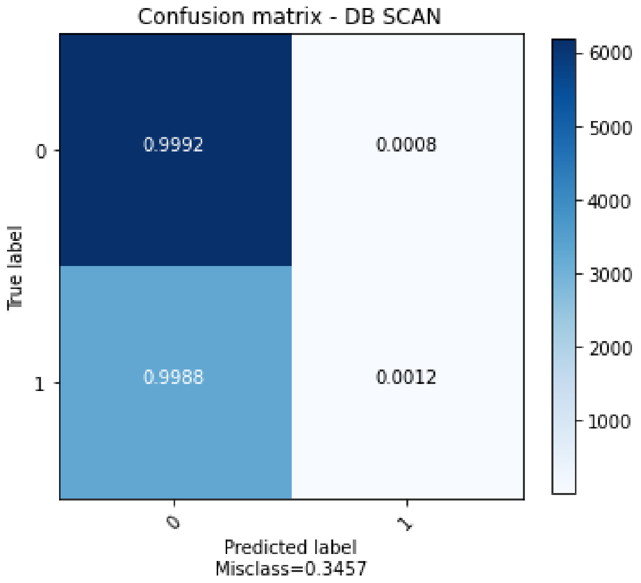
Confusion matrix - Mean shift

Misclass=0.3456









A paragraph explaining which of your Unsupervised Learning models you recommend as a final model that best fits your needs in terms.

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

1. K-means has the highest misclassification error; hence it is clustering worst.
2. Even though mean shift algorithm has less error, it has zero silhouette score which means it is grouping all points into one cluster which is wrong since we know we have 2 clusters.
3. Complete and Ward agglomerative clustering has low accuracy and high silhouette score whereas Single linkage clustering and DBSCAN has high accuracy and lower silhouette score.

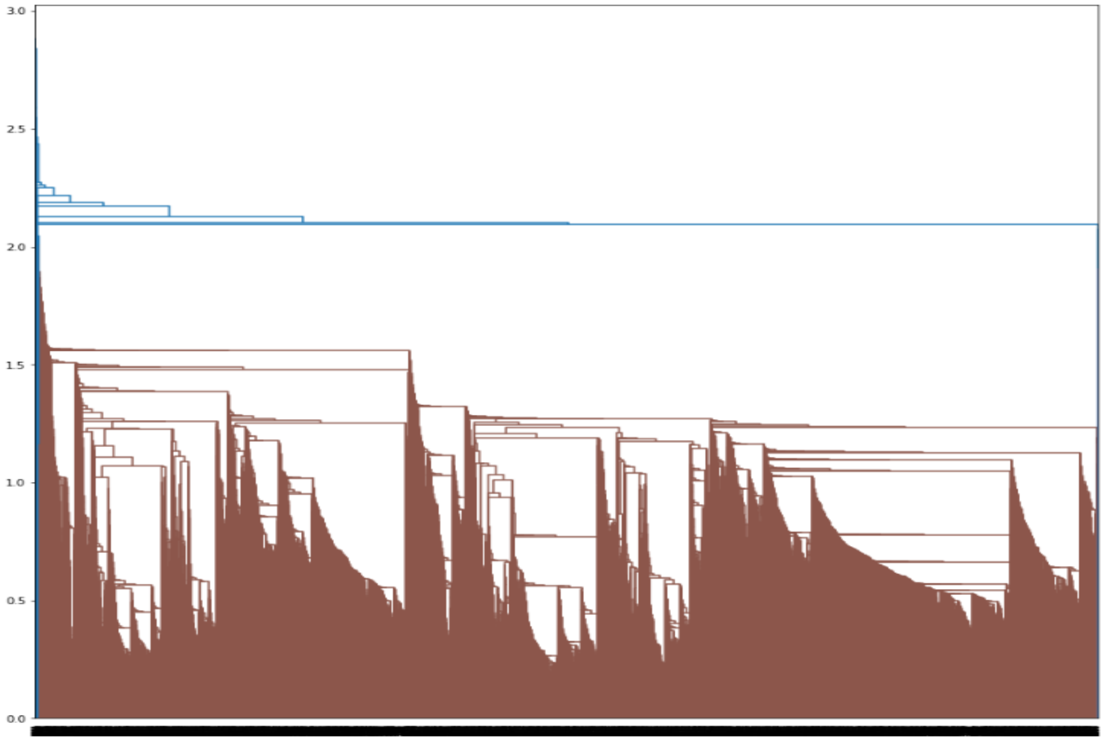
Since we prefer accuracy more than silhouette score, I am now comparing Single linkage clustering and DBSCAN methods.

Even though they have very similar accuracy and misclassification, Single linkage clustering gives higher silhouette score and from an explainability point of view also, it is better.

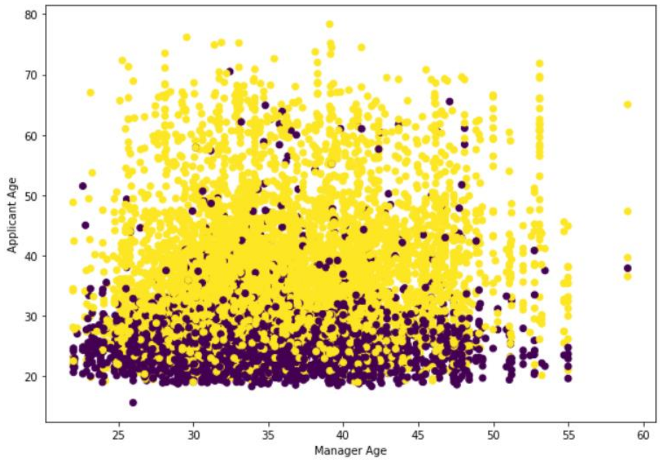
Hence, I will recommend to use Agglomerative hierarchical clustering with single linkage with parameters as mentioned above.

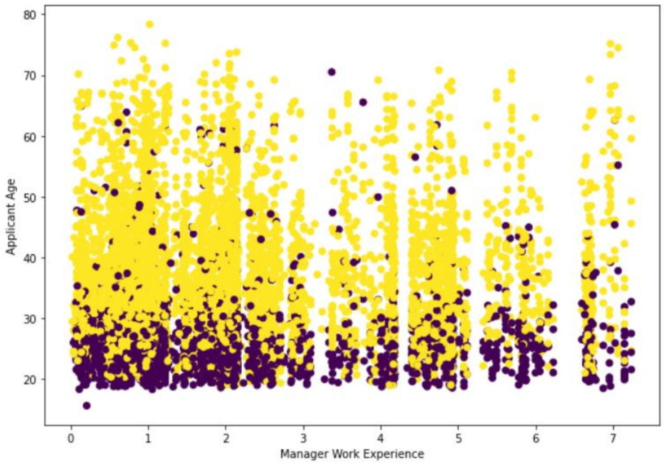
Summary Key Findings and Insights, which walks your reader through the main findings of your modeling exercise.

Dendrogram for the final model is:



To see effect of clustering I have plotted few important variables which came out from logistic regression analysis, colour coded by clustering algorithm prediction.





Findings:

1. Single linkage clustering gives decent performance.
2. We can see overall cluster boundaries from a few important variables like Applicant age, manager age and manager's work experience.
3. Also, the misclassification points tend to be in interior of the larger cluster although the above plots don't convey the entire picture since the data in more than 3 dimensional.

Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model or adding specific data features to achieve a better model.

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.

The current model does not have high accuracy hence, some more hyper parameter tuning needs to be done.

Doing PCA and applying clustering on top of that to improve performance is a possible next step. Implementing OPTICS algo which is improved DBSCAN can be done as a next step. When I did it directly the problem is number of clusters is not a parameter hence, we cannot fix number of clusters. So, some clubbing of clusters is needed before final evaluation.