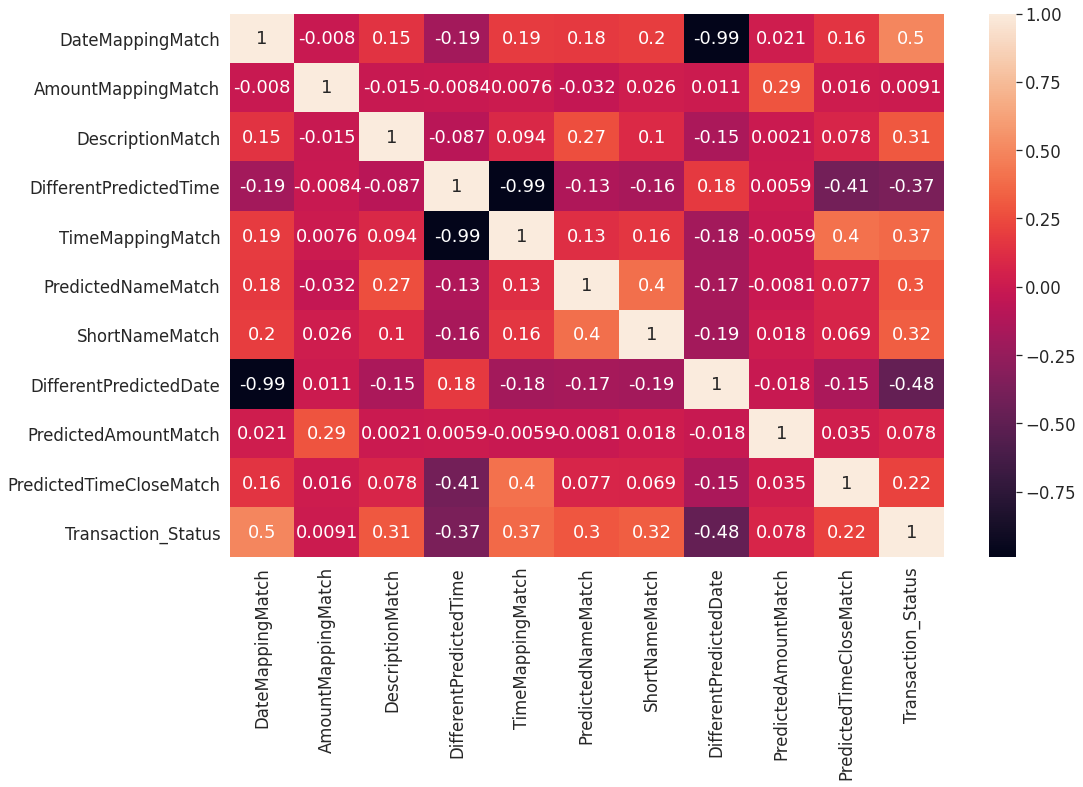
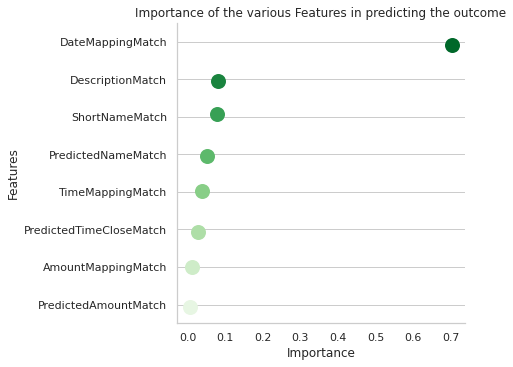
# Approach:

1. Importing the libraries and data.
2. Data Preprocessing
   * Understanding type of features in the data (Ex: Categorical/Continuous)
   * Creating the Labelled column from the data (Arriving at Transaction\_Status which is target variable achieved using matched\_transaction\_id and feature\_transaction\_id)
   * Drop the irrelevant features (matched\_transaction\_id and feature\_transaction\_id)
3. Exploratory Data Analysis (EDA)
   * Visualize the target feature to understand the class imbalance.
   * Correlation Matrix to understand correlations of one feature with others in the dataset.
   * Drop the less important features (Based on correlation matrix score of features with the target variable, DifferentPredictedDate and DifferentPredictedTime were having the least score.)
4. Balancing the DataSet using SMOTE
   * Transform the dataset to respond to class imbalance. (Instead of using sampling techniques like oversampling and undersampling which helps in balancing out data but does not add any new information to the model, hence used SMOTE which is a type of data augmentation for minority class)
5. Train and Test split Dataset.
6. Hypermater tuning with multiple Models.
   * Created a utility function using GridSearchCV with a CV score of 10 to run ML models with different combinations of hyper parameters. (Since the dataset is small used GridSearchCV to run on all the combinations of hyperparameters passed to the models, else if the dataset is huge we can consider using RandomisedSearchCV)
   * Run different models and analysed the classification report to understand precision/recall/f1-score. (Since it is an imbalance dataset we can avoid using accuracy as performance evaluation metric for model and use precision/recall/f1Score)
7. Save the model with better evaluations in terms of precision/recall/f1Score.

**Correlation Matrix Information:**



**Important Features Information:**

****

# Results:

| **Model** | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| --- | --- | --- | --- | --- |
| **Random Forest Classifier** | 0.93 | 0.92 | 0.92 | 0.9687 |
| **Gradient Boosting** | 0.93 | 0.92 | 0.92 | 0.9684 |
| **K Nearest Neighbors** | 0.93 | 0.92 | 0.92 | 0.9655 |
| **Adaptive Boosting** | 0.93 | 0.91 | 0.91 | 0.9644 |
| **Logistic Regression** | 0.92 | 0.91 | 0.91 | 0.9623 |
| **Support Vector Machine** | 0.93 | 0.91 | 0.91 | 0.9436 |
| **ANN** | 0.93 | 0.92 | 0.92 | 0.9177 |

Ensemble models provided better results in terms of Precision/Recall/F1Score and Accuracy (Considered since we have balanced the dataset.). On basis all the metrics RandomForestClassifier stayed top in terms of all the metrics, hence considered that as the model and saved the model for the same to leverage it on the test set.

# Recommendations:

1. Data acquisition can be improved, since we are using synthesised dataset to overcome the imbalance in the dataset (current we see a split of 92.8% in one class and 7.2% in another class), results could be deviating to some extent but where in if we have larger data we can avoid this sampling technique to improvise the model
2. Information on features can be provided that can lead to understanding and engineering multiple features thus reducing the dimensionality of the dataset.