CS 6923 Machine Learning
Spring 2019
Final Project Report

Name: Hardik Jivani

NetID: haj272

Name: Virendra Singh Rajpurohit

NetID: vsr266

PART I: Preprocessing (No more than two pages for this part)

1. **How does your program handle missing value? And why?**

- We are using SimpleImputer from sklearn.impute for imputation of missing values
- For 'House' feature, we used the "constant" strategy with fill_value='Other'. Since, we can put 'Other' as the category for the missing/NaN ~2000 values
- For 'Move_specialty' and 'Player_code' features, we used the "most_frequent" strategy as we assumed missing/Nan values are almost half of the total data. ~50000
- For 'weight', we had to drop the whole column as it is not providing tangible information and does not contribute to predict our target variable.
- We used both training and test data for handling missing values.

2. **If your program converts numeric features to categorical features, or categorical features to numeric features. Describe how it does it.**

- Categorical features to Numeric features - 'house', 'gender', 'player_code', 'move_specialty', 'player_type', 'snitchnip', 'stooging', 'change', 'snitch_caught'
- We used Factorize method of Pandas to convert each of the category in the categorical feature to numeric values (say for snitchnip : None, Norm, High gets converted to 0,1,2)
- For each feature, it is collecting unique categories to represent in the integer values.
- For nominal features such as 'foul_type_id','game_move_id','penalty_id', we factorized them to get the numeric features. Example 'foul_type_id' feature contains categories represented in numbers. Thus, to convert the numeric categories to numeric data we used factorize method of pandas library.
- We used Binary Encoder method for these features as the numeric data in classification problem suggests importance to higher numeric data than lower which should not happen. Thus, we converted the numeric data to binary data representation
- Since, One-hot encoding was increasing the number of columns drastically, we used Binary encoder which has a lot fewer dimensions comparatively.
- We used both training and test data for feature conversion.

3. **Describe any feature selection, combination or creation, and any feature values combination performed by your program and the reasons for doing so.**

- More number of features and more categories in a feature means increase in dimensions of search space for the problem. Some features may not be important and can end up overfitting out model. We can combine features or simply eliminate features which has high correlation with each other.
- Features such as 'double_eight_loop' and 'finbourgh_flick' consist only one unique category i.e. “No”, which doesn’t help in fitting better model. So, we dropped the features.
- There were total 23 tactics which are being associated to each player. Each tactic is associated with some changes such as No change, steady change, gone up or gone down. We reduced the 23 tactic features to 4 features such as "No", "Steady", "Up" and "Down" by representing one where the player had that tactic and 0 otherwise.
- Features such as 'num_games_notpartof', 'num_games_injured' and 'num_games_satout' contains similar context of being not taking part in the game. So, we reduced all the three features to only one by getting the sum of all the corresponding values and called it as ‘num_games_notpartof’
- We reduced the number of categories in snitchnip from 4 to 3. We merged two categories ‘>300’ and ‘>200’ to a single category ‘High’
- Similarly, category reduction in stooging from 4 to 3. We merged two categories ‘>7’ and ‘>8’ to a single category ‘High’
- Since, ‘id_num’ and ‘player_id’ are highly correlated and these two features are not helpful in fitting better model because they are specific to each player. We dropped the features from the training as well as test dataset.
- We used training and test dataset for feature creations and feature reduction.

4. **Describe other preprocessing used in your program(e.g. centralizing, normalization?**

- Train data comprises of attributes with varying scale. Thus to rescale out data, we can use normalization, standardization etc. We used standardization, to centre out data with 0 mean and eliminate data points which are 3SD away. This will scale our data to handle varying/extreme data points. We fit the StandardScaler model using the training dataset and transformed training and test dataset.
- Outliers are the extreme values which diverges from the overall pattern. To remove outliers, we eliminated rows that had data points above (mean + 3SD) and points below (mean - 3SD) for each of the features(containing numeric data) in ['age','num_game_moves','num_game_losses','game_duration','num_practice_sessions','num_games_notpartof','num_games_won','No','Steady','Up','Down'].
- Outliers were being removed only from the training dataset.

PART II: Classification (No more than two pages for each model in this part)
Note: Values for Precision, Recall, F1_score are for class 1/"YES"
Model One:

1. **Supervised learning method used in this model is**
   Logistic Regression using SMOTE re-sampling method

2. **Why you choose this supervised learning method?**

- Logistic regression is a very efficient and easy technique to implement for binary classification as its outcome is discrete and can be used as a performance baseline.
- It does not require too many computational resources and is easy to regularize.
- It measures the relationship between our target variable and other independent features by estimating probabilities using underlying logistic function.
- The probabilities are then transformed to 0,1 with threshold classifier and do the prediction.
- Since, feature optimization increases the performance drastically, we chose logistic regression for a problem that has a lot of features.

3. **Describe the method you used to evaluate this method.**

- We used precision, recall and F1 score for evaluation of the problem.
- Precision is the ratio of correctly predicted observations to the total predicted positive observations.
- Recall is the ratio of correctly predicted observations to the total observations in the actual class.
- F1 score is the harmonic mean of precision and recall.
- This evaluation (F1 Score) takes false positives and false negatives into account which is ideal for highly imbalanced dataset like this one.
- For this dataset, F1 score is more useful than accuracy.
- Since, the cost of false positive and false negative are different, it is better to use F1 score as the evaluation.
- Higher F1 score means better prediction.

4. **Describe process of experimenting different parameter settings or associated techniques.**

- Parameter name: C (Inverse of Regularization strength)
  - Parameter values: [1.00000000e-05, 1.77827941e-02, 3.16227766e+01, 5.62341325e+04, 1.00000000e+08]
  - Performance of different values:
    - Value=1.00000000e-05 (Precision:0.0, Recall:0.0, F1-score:0.0)

- Value=1.77827941e-02, 3.16227766e+01, 5.62341325e+04, 1.00000000e+08 (Precision:0.17, Recall:0.0, F1-score:0.0)

- Analysis:
  - For C parameter, F1 score is not varying but precision is increasing which suggests that more number of predictions are correct as compared to the training set.
  - Since, the recall is close to zero, means that very few results are returned. Thus, we can use the higher value of C. We limit C because higher C would overfit the data.
  - Most Suitable C=0.01

○ Parameter name: Sampling Strategy (SMOTE)

- Parameter values: 0.4, 0.6, 0.75, 1.0
- Performance of different values:
  - Value=0.4 (Precision:0.20, Recall:0.04, F1-score:0.07)
  - Value=0.6 (Precision:0.18, Recall:0.17, F1-score:0.17)
  - Value=0.75 (Precision:0.16, Recall:0.31, F1-score:0.21)
  - Value=1 (Precision:0.14, Recall:0.55, F1-score:0.23)
- Analysis:
  - For Sampling strategy of SMOTE, we are changing the number of synthetic samples keeping C value higher i.e 0.01.
  - From the results, we can see that as number of synthetic samples increases, recall and F1 score increases suggesting a better model, but sampling 1.0 is impractical as that would perfectly balance the class distribution. Thus limiting ratio to 0.75
  - At SS=1.0, we get the highest F1 score but since precision is decreasing we limit SS to 0.75
  - *Most Suitable SS=0.75*

4. Accuracy and Confusion matrix with most suitable parameters.

		Predicted	
		Yes	No
Correct	Yes	778	1770
	No	3959	17001

Accuracy = `0.756295729113`

Model Two:

1. **Supervised learning method used in this model is**

   Decision Tree Classifier

2. **Why you choose this supervised learning method?**
   Decision tree are white boxes. The outcomes are transparent and can be easily interpreted. It can be applied for both numeric and categorical features. In fact, it performs better than other algorithms when the dataset has categorical features. When we fit a decision tree to a training set, the tree is split with the most important feature, thus feature selection is automatic. This algorithm is not prone to outliers since the splitting is not based on absolute values but on proportion of samples within the split ranges. It also doesn't require assumptions of linearity of the data and hence we use this algorithm to get better evaluation even if data has some non-linear dependency as well as categorical features.

3. **Describe the method you used to evaluate this method.**

- We used precision, recall and F1 score for evaluation of the problem.
- Precision is the ratio of correctly predicted observations to the total predicted positive observations.
- Recall is the ratio of correctly predicted observations to the total observations in the actual class.
- F1 score is the harmonic mean of precision and recall.
- This evaluation (F1 Score) takes false positives and false negatives into account which is ideal for highly imbalanced dataset like this one.
- For this dataset, F1 score is more useful than accuracy.
- Since, the cost of false positive and false negative are different, it is better to use F1 score as the evaluation.
- Higher F1 score means better prediction.

4. **Describe process of experimenting different parameter settings or associated techniques.**

- Parameter name: Max_features
  - Parameter values: 10, 25, 40, 52
  - Performance of different values:
    - Value=10 (Precision:0.19, Recall:0.04, F1-score:0.06)
    - Value=25 (Precision:0.15, Recall:0.06, F1-score:0.09)
    - Value=40 (Precision:0.16, Recall:0.08, F1-score:0.10)
    - Value=52 (Precision:0.18, Recall:0.09, F1-score:0.12)
  - Analysis:
    - *Max_features is the number of features to consider when looking for the best feature. Increasing Max_features will definitely increases our performance as number of features are too many for our problem. It suggests that the model is looking for the best feature in the whole feature space (Max_features=52) and produces higher performance.*
    - *Most suitable Max_features=52*
- Parameter name: Sampling Strategy
  - Parameter values: 0.4, 0.5, 0.6, 1
  - Performance of different values:
    - Value=0.4 (Precision:0.12, Recall:0.15, F1-score:0.14)
    - Value=0.5 (Precision:0.15, Recall:0.18, F1-score:0.16)
    - Value=0.75 (Precision:0.14, Recall:0.19, F1-score:0.16)
    - Value=1 (Precision:0.14, Recall:0.18, F1-score:0.16)
  - Analysis:
    - *For sampling strategy of SMOTE with higher synthetic samples improves the performance as we have imbalance dataset. We get F1 score = 0.16, when we have perfectly balanced class distribution for both 'Yes' and 'No'. But the F1 or precision or recall doesn't seem to vary much after 0.6 so we can keep the SS to be 0.75 which produces F1=0.16*
    - *Most Suitable SS=0.75*

4. **Accuracy and Confusion matrix with most suitable parameters**

		Predicted	
		Yes	No
Correct	Yes	441	2170
	No	2718	18242

Accuracy: `0.794750723158`

Model Three:

1. **Supervised learning method used in this model is**

   Multi-Layer Perceptron Classifier

2. **Why you choose this supervised learning method?**
   MLP can better model data with non-constant variance and high volatility. It has ability to learn hidden relationships in the data without assuming any fixed relationship in the data. It does not make any assumption regarding any probabilistic information about the pattern classes under consideration or underlying probability density functions in comparison to other models. It can yield the required decision function directly via training. We chose MLP as it can be used to extract patterns and detect trends that are too complex to be noticed by other models.

3. **Describe the method you used to evaluate this method.**

- We used precision, recall and F1 score for evaluation of the problem.
- Precision is the ratio of correctly predicted observations to the total predicted positive observations.
- Recall is the ratio of correctly predicted observations to the total observations in the actual class.
- F1 score is the harmonic mean of precision and recall.
- This evaluation (F1 Score) takes false positives and false negatives into account which is ideal for highly imbalanced dataset like this one.
- For this dataset, F1 score is more useful than accuracy.
- Since, the cost of false positive and false negative are different, it is better to use F1 score as the evaluation.
- Higher F1 score means better prediction.

4. **Describe process of experimenting different parameter settings or associated techniques.**

- Parameter name: Sampling Strategy
  - Parameter values: 0.4,0.6,0.75,1
  - Performance of different values:
    - Value=0.4 (Precision:0.26, Recall:0.03, F1-score:0.06)
    - Value=0.6 (Precision:0.18, Recall:0.21, F1-score:0.19)
    - Value=0.75 (Precision:0.16, Recall:0.31, F1-score:0.21)
    - Value=1 (Precision:0.1, Recall:0.34, F1-score:0.23)
  - Analysis:
    - For sampling strategy of SMOTE with higher synthetic samples improves the performance as we have imbalance dataset.
    - We get F1 score = 0.16, when we have even class samples (SS=1) for both 'Yes' and 'No'.
    - We limit SS to 0.75 to have a more practical approach for imbalance dataset.
    - The precision is better at 0.75 comparatively thus data is predicted more accurately. Most suitable SS=0.75
- Parameter name: Hidden_layer_sizes
  - Parameter values: (120),(120,60),(120,120,120)
  - Performance of different values:
    - For all values= (Precision:0.0, Recall:0.0, F1-score:0.0)
  - Analysis:
    - Precision, Recall, F1-score does not seem to vary while changing the number of layers or number of nodes in one layer.

5. **Accuracy and Confusion matrix with most suitable parameters**

		Predicted	
		Yes	No
Correct	Yes	891	1657
	No	4401	16559

Accuracy: `0.742300493449`

PART III: Best Hypothesis (No more than two pages for this part)

1. **Which model do you choose as final method?**
   Model number: 1
   Supervised learning method used in this model: Logistic Regression with SMOTE

2. **Reasons for choosing this model.**

- Logistic regression is a very efficient and easy technique to implement for binary classification as its outcome is discrete and can be used as a performance baseline.
- This model works the best since we have a lot of features.
- When re-sampled dataset using SMOTE, it produces better performance compared to other models.
- F1 score at SS=0.75 and C=0.01 has a high value of 0.21 with fair number of correct classification of 16% and fair number of outputs returned i.e 31%.
- These results are similar to MLP but logistic runs way faster than MLP.
- Thus, we chose Logistic regression with SMOTE.

3. **What are the reasons do you think that make it has the best performance?**

- By re-sampling imbalanced dataset and by choosing the right machine learning algorithm we can improve the prediction performance for minority class. With this model, we are achieving, higher F1 score (high precision and high recall) for the minority class.
- Although, we get high accuracy for other models, but higher accuracy is representing the underlying imbalanced class distribution. Thus, we do the evaluation by checking higher F1 score for SMOTE implementation
- Logistic regression tries the minimalistic approach of fitting the hypothesis that goes through a sigmoid classifier. Its better performance suggest that the input variables are highly relational to the target variables. Since, most of the 'important' features are numeric in nature, thus logistic regression seems to be the best model for this classification problem.
- For the large dataset, centralizing the data to mean and feature optimization creates the best platform for logistic to run efficiently which reflects in the evaluation.