IE581 Project
Final Presentation
Bank Customer Churn Prediction

Emre Eren, M. Furkan Oruc

Overview

1. Overview of the Dataset
2. Preprocessing
3. Feature Selection
4. Evaluation Metrics
5. Random Forest & KNN Implementation
6. Ensemble Algorithms
7. Decision Tree & Naive Bayes with Random Undersampling by Observation
8. Interpretation of Decision Tree
9. Comparison

Data Preprocessing

Churn	Not-Churn
2099	27901

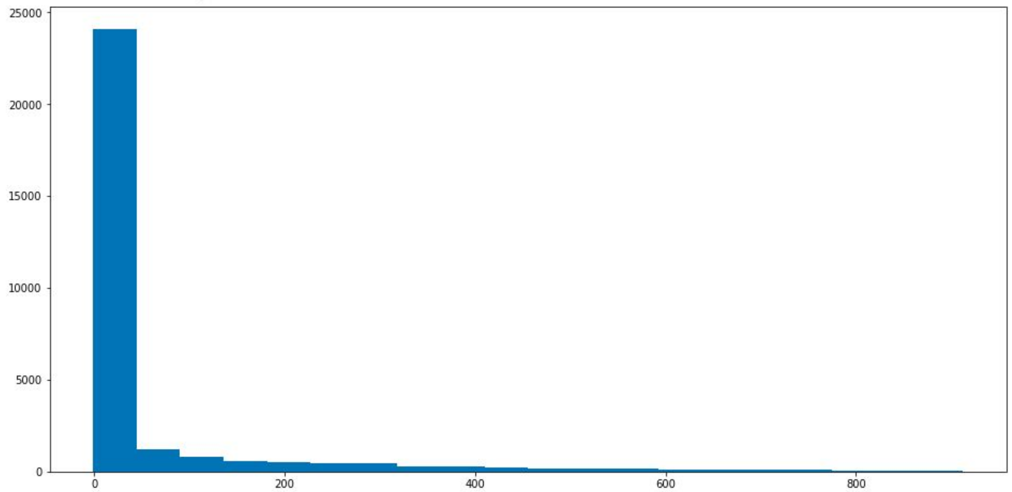
Churn	Not-Churn
2077	27872

After dropping rows in which <0 values are heavily found

```
df.drop(df[df['DEBIT_LOGIN_GECEN_SURE'] < 0].index, inplace=True)
df.drop(df[df['ATM_FIN_ISLEM_GECEN_SURE'] < 0].index, inplace=True)
df.drop(df[df['CM_LOGIN_GECEN_SURE'] < 0].index, inplace=True)
```

Data Preprocessing

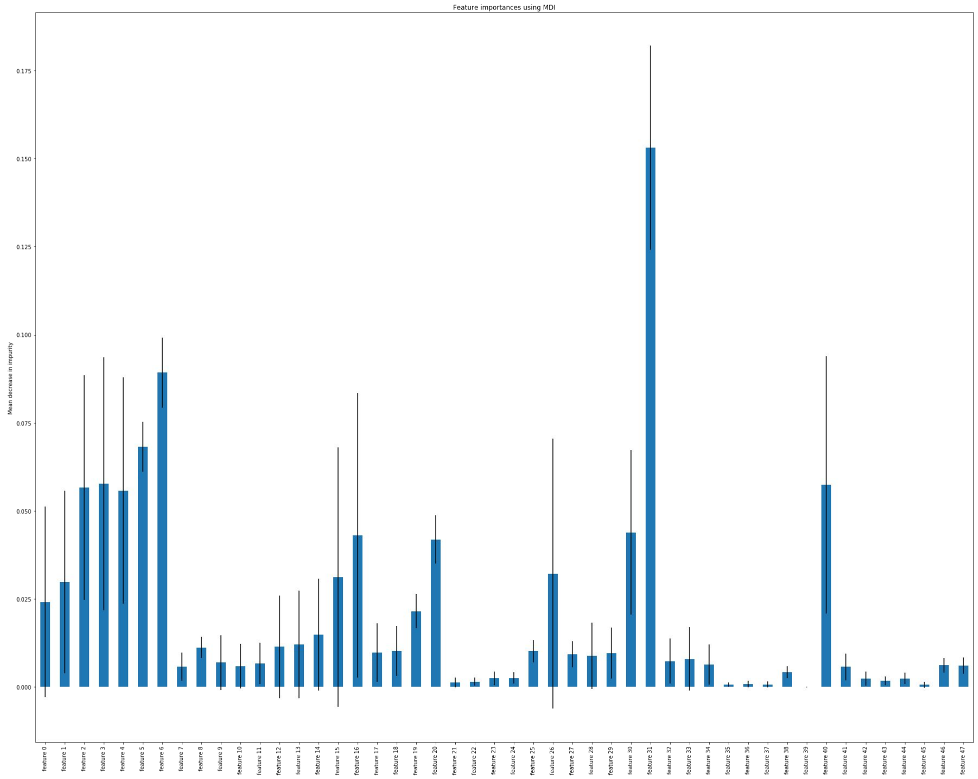
- Percentile based categorization is conducted on skewed data.
- During iterations of preprocessing, manually observations are conducted for best categorical split.
- Categorizations are conducted based on visual observations.
- By means of dummy creation methodology, new columns are generated, which include 4 different percentile populations of each columns.
- Null values are filled with (-1) to seperate them as a distinct category for the algorithm to interpret.



DEBIT_FIN_ISLEM_GECEN_SURE

Feature Importance Tests

1. Random Forest Classifier based importance test: Parameter is the Mean Decreased Impurity
   a. Computed as the mean and standard deviation of accumulation of the impurity decrease within each tree.



2. Permutation Based importance test (Random Forest Based)

Selected Features to Include in the Model

0	DEBIT_FIN_ISLEM_GECEN_SURE	29949	non-null	float64
1	DEBIT_LOGIN_GECEN_SURE	29949	non-null	float64
2	VDSZ_BKYORT_Ilk3	29949	non-null	float64
3	VDSZ_BKYORT_Ikinci3	29949	non-null	float64
4	VDSZ_BKYORT_Ucuncu3	29949	non-null	float64
5	MUSTERILIK_YASI	29949	non-null	float64
6	MUSTERI_YASI	29949	non-null	float64
7	VDSZ_SHPLK_FLAG	29949	non-null	int64
8	MUS_CLSYRM_FLAG	29949	non-null	float64
9	ATM_ORT_Ilk3	29949	non-null	float64
10	ATM_ORT_Ikinci3	29949	non-null	float64
11	ATM_ORT_Ucuncu3	29949	non-null	float64
12	ATM_LOGIN_GECEN_SURE	29949	non-null	float64
13	ATM_FIN_ISLEM_GECEN_SURE	29949	non-null	float64
14	CM_LOGIN_GECEN_SURE	29949	non-null	float64
15	SUBE_FIN_ISLEM_GECEN_SURE	29949	non-null	float64
16	ATM_TERK_TAR_GECENSURE	29949	non-null	float64
17	SUBE_TERK_TAR_GECENSURE	29949	non-null	float64
18	VDSZ_TERK_TAR_GECENSURE	29949	non-null	float64
19	GUNCEL_SEGMENT_IKT	29949	non-null	uint8

Evaluation Metrics in the Scope of **Class Imbalance**

		Prediction	
	Total Population	TRUE	FALSE
Condition	TRUE	True Positive	False Positive
	FALSE	False Negative	True Negative

$$\frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive + False\ Positive}} = \boldsymbol{Precision}$$

$$\frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Postive + False\ Negative}} = \boldsymbol{Recall}$$

- **Roc-Auc** score indicates how good each class is separated within each other as a result of the classification. (Gareth James, 2017) It's a significant method since it's one of the least biased evaluation metrics to imbalance. (Ali, 2013)

Random Forest & K.N.N. & Logistic Regression Implementation

1. Random Forest Classifier has been the best performer among others.

```
Roc_Auc_forest
```

0.7490196697459799

```
Accuracy_forest
```

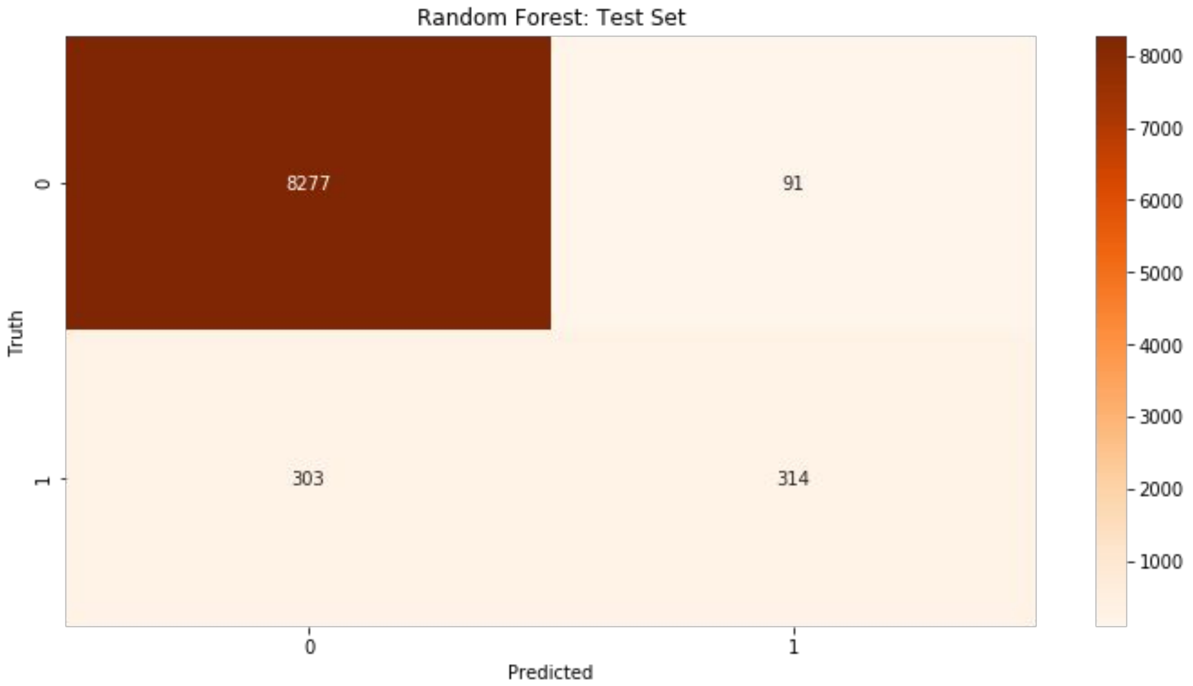
0.9561491374513077

```
Precision_forest
```

0.7753086419753087

```
Recall_forest
```

0.5089141004862237



Decision Tree and Logistic Regression based Ensemble Models

1. 10 Fold Stratified Cross Validation has been performed.
2. Random Undersampling based two meta-algorithms are applied.
3. Decision Tree based EasyEnsemble performed slightly better than others.

	EasyEnsemble DT	RUSBoost DT	RUSBoost LR
Mean Precision Score, 10 Fold	0.421176	0.412421	0.357798
Mean Recall Score, 10 Fold	0.894537	0.887361	0.889267
Mean ROC-AUC Score, 10 Fold	0.901362	0.896536	0.885112
Mean Accuracy Score, 10 Fold	0.907242	0.904438	0.881532

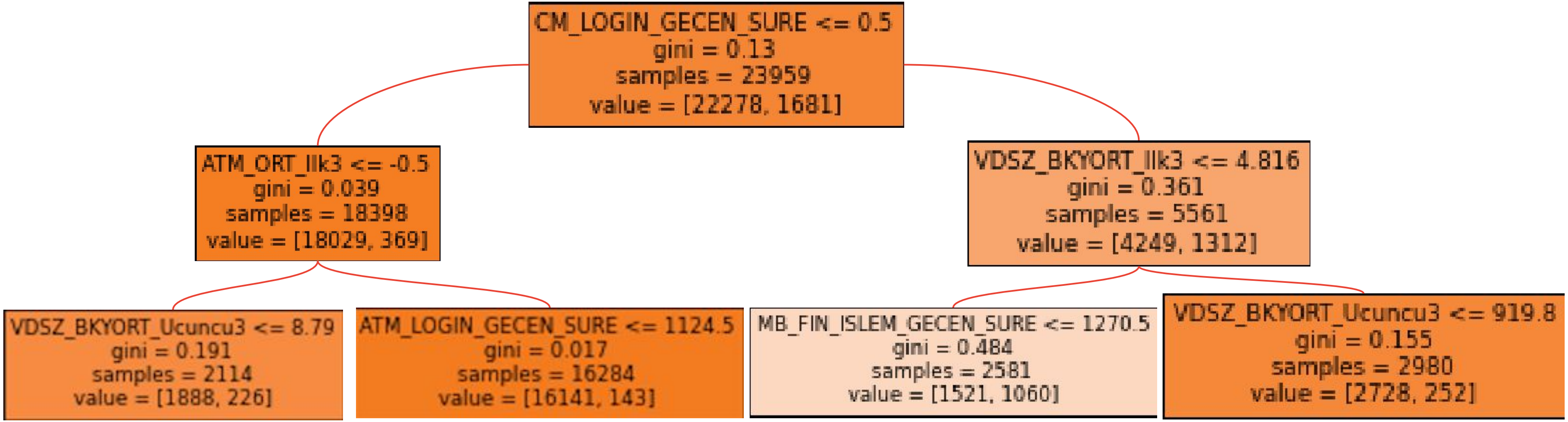
Naive Bayes & Decision Tree with Random Undersampling

Decision Tree and Naive Bayes are compared with their respective results in two versions: original class distribution and manually ensembled (0.2). Result can be observed below.

	Naive Bayes	N.B. R. Undersampled	Decision Tree	D.T. R. Undersampled
Precision	0.094753	0.099279	0.688525	0.547059
Recall	0.916667	0.936791	0.476499	0.753647
Accuracy	0.415526	0.412020	0.949249	0.940234
Roc_Auc	0.648358	0.655059	0.730303	0.853819

Decision Tree Results: Business Interpretation

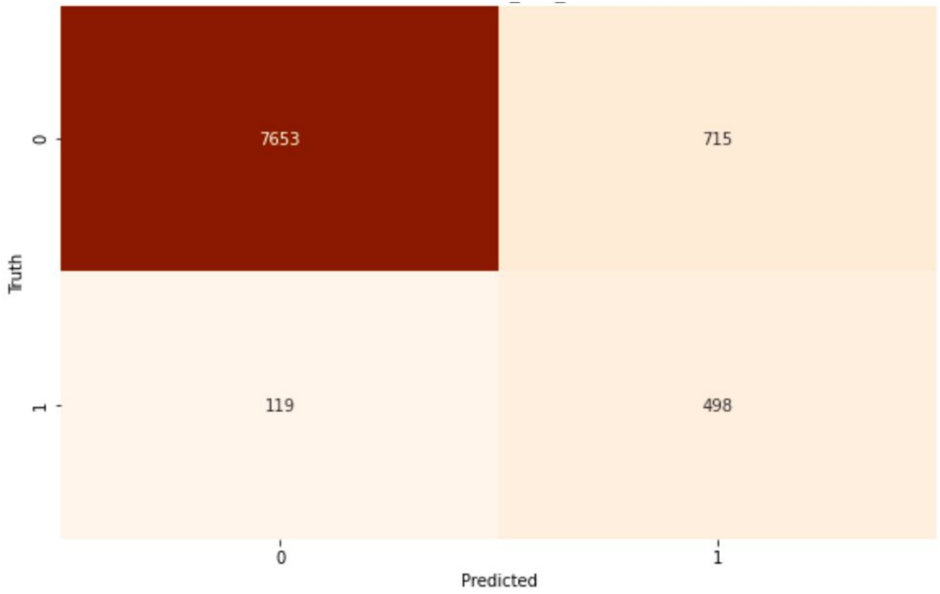
Based on Gini Index based impurity analysis of decision tree, CM_LOGIN_GECEN_SURE and VDSZ_BKYORT_Ilk3 have been some of the most determinative features, along with ATM_ORT_Ilk3.



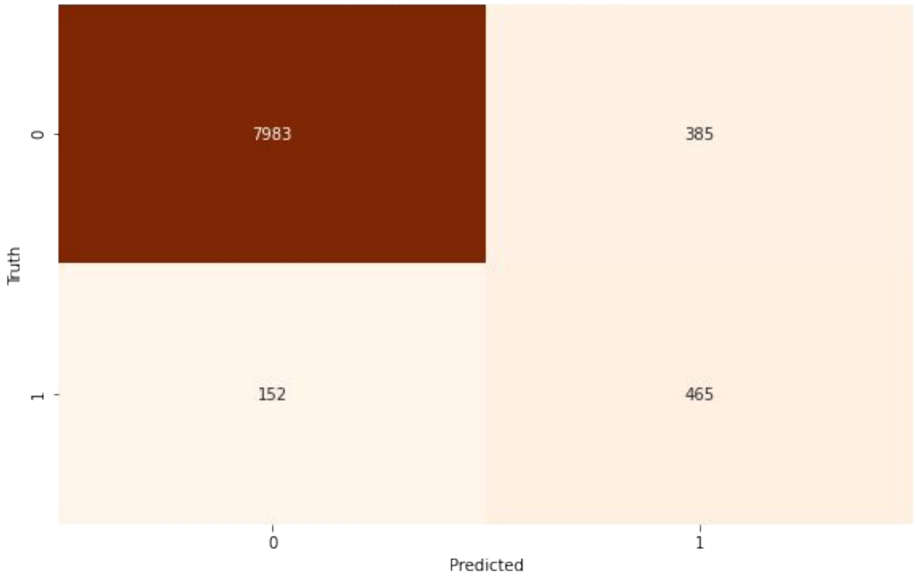
Comparison between **best Performers**

Manually conducted random undersampling based decision tree and EasyEnsemble based Decision Tree algorithms are compared. Manually ensembled algorithm slightly outperforms the ensemble.

Decision Tree Based EasyEnsemble



Random Undersampling [0.2] & Decision Tree



Comparison between **best Performers**

Manually conducted random undersampling based decision tree and EasyEnsemble based Decision Tree algorithms are compared. Manually ensembled algorithm slightly outperforms the ensemble.

Decision Tree Based EasyEnsemble **Random Undersampling [0.2] & Decision Tree**

	EasyEnsemble DT	D.T. R. Undersampled
Mean Precision Score, 10 Fold	0.421176	0.547059
Mean Recall Score, 10 Fold	0.894537	0.753647
Mean ROC-AUC Score, 10 Fold	0.901362	0.940234
Mean Accuracy Score, 10 Fold	0.907242	0.853819

Thank you.

Imbalanced Data, Review

- When a class outnumbers another class in a dataset, traditional machine learning algorithms are challenged in various ways.

- Algorithms such as Backpropagation Neural Networks, Decision Trees and KNN are some of the prominent ones which may not identify the minority class member instances in the most precise way. (Ali, 2015)

- For an algorithm to be prone to imbalance is also driven by linear separability of a dataset as well. Linearly separable datasets are not that sensitive to imbalance as much as higher complexity degrees. (Rekha, 2019)

- Skewed data distribution is the most common observed class imbalance prevalence. On the other hand, small sample size and existence of within subclass concepts are other most prominent imbalance challenges. (Ali, 2015)

- Fraud Detection, Manufacturing Faults, Detection of Oil Spills and Medical Diagnosis are some of the prominent research areas suffering from class imbalance.

Overview of Imbalance Focused Solutions

- Namely, solutions addressing class imbalance issue is categorized under three subjects, data level, algorithm level and ensemble (hybrid).

- Data Level
    - Class imbalance is addressed via either sampling methods or oversampling and undersampling the minority and majority classes, respectively. RUS (Random Undersampling) and SMOTE are some of the examples. (Rekha, 2019)
    - A potential drawback for these approaches can be mentioned as decreased computational efficiency due to increased number of samples or loss of information rich instances due to undersampling.
- Algorithm Level
    - Modification of existing algorithms by producing either new parameters or creation of new approaches can be mentioned.
- Ensemble (Hybrid)
    - Ensemble methods combine data level and algorithm level approaches, such as bagging together with oversampling or undersampling. Most recent developments in the domain to improve the performance are based on ensembles. Some are AdaBoost, RusBoost and EasyEnsemble. (Rekha, 2019)

References

- T. Liu, "EasyEnsemble and Feature Selection for Imbalance Data Sets," 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Shanghai, 2009, pp. 517-520, doi: 10.1109/IJCBS.2009.22.
- Ali, Aida & Shamsuddin, Siti Mariyam & Ralescu, Anca. (2015). Classification with class imbalance problem: A review. 7. 176-204.
- Rekha, Gillala & Reddy, V & Tyagi, Amit. (2019). A novel approach for solving skewed classification problem using cluster based ensemble method. Mathematical Foundations of Computing. 3. 10.3934/mfc.2020001.
- T. Chengsheng, L. Huacheng, and X. Bing, “Adaboost typical algorithm and its application research,” MATEC Web of Conferences, vol. 139, p. 00222, 01 2017.
- Xiao, Han & Rasul, Kashif & Vollgraf, Roland. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.
- M. Zikeba, S. K. Tomczak, and J. M. Tomczak, “Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction,” Expert Systems with Applications, 2016.
- C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “Rusboost: A hybrid approach to alleviating class imbalance,” IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 40, no. 1, pp. 185–197, 2010.

Appendix

```
Accuracy_logistic
```

```
0.9307735114079021
```

```
Recall_logistic
```

```
0.009724473257698542
```

```
Precision_logistic
```

```
0.35294117647058826
```

```
Conf_logistic
```

```
array([[8357,   11],
       [ 611,    6]])
```

```
Accuracy_KNN
```

```
[0.9434613244296048,
 0.9475792988313857,
 0.9462437395659432,
 0.9483583750695603,
 0.9469115191986645,
 0.9492487479131887,
 0.9475792988313857,
 0.9483583750695603,
 0.9462437395659432]
```

```
Precision_KNN
```

```
[0.6224719101123596,
 0.737012987012987,
 0.6700507614213198,
 0.7593220338983051,
 0.6955307262569832,
 0.7692307692307693,
 0.7085714285714285,
 0.7647058823529411,
 0.6982248520710059]
```

```
Recall_KNN
```

```
[0.44894651539708263,
 0.3679092382495948,
 0.42787682333873583,
 0.36304700162074555,
 0.4035656401944895,
 0.3727714748784441,
 0.4019448946515397,
 0.3581847649918963,
 0.3824959481361426]
```