Loan Prediction

By Adriana Caetano and Torsha Mazumdar

Motivation

Dream Housing Finance company deals in all home loans

Automate loan eligibility

Online application form

Problem Category

Supervised Learning - learning from the data based on sample input-output pairs.

- **Input** - Customer Attributes, Loan Amount, Term
- **Output/Label** - Loan Status

Binary Classification - Loan Status is either Y(Yes) or N(No).

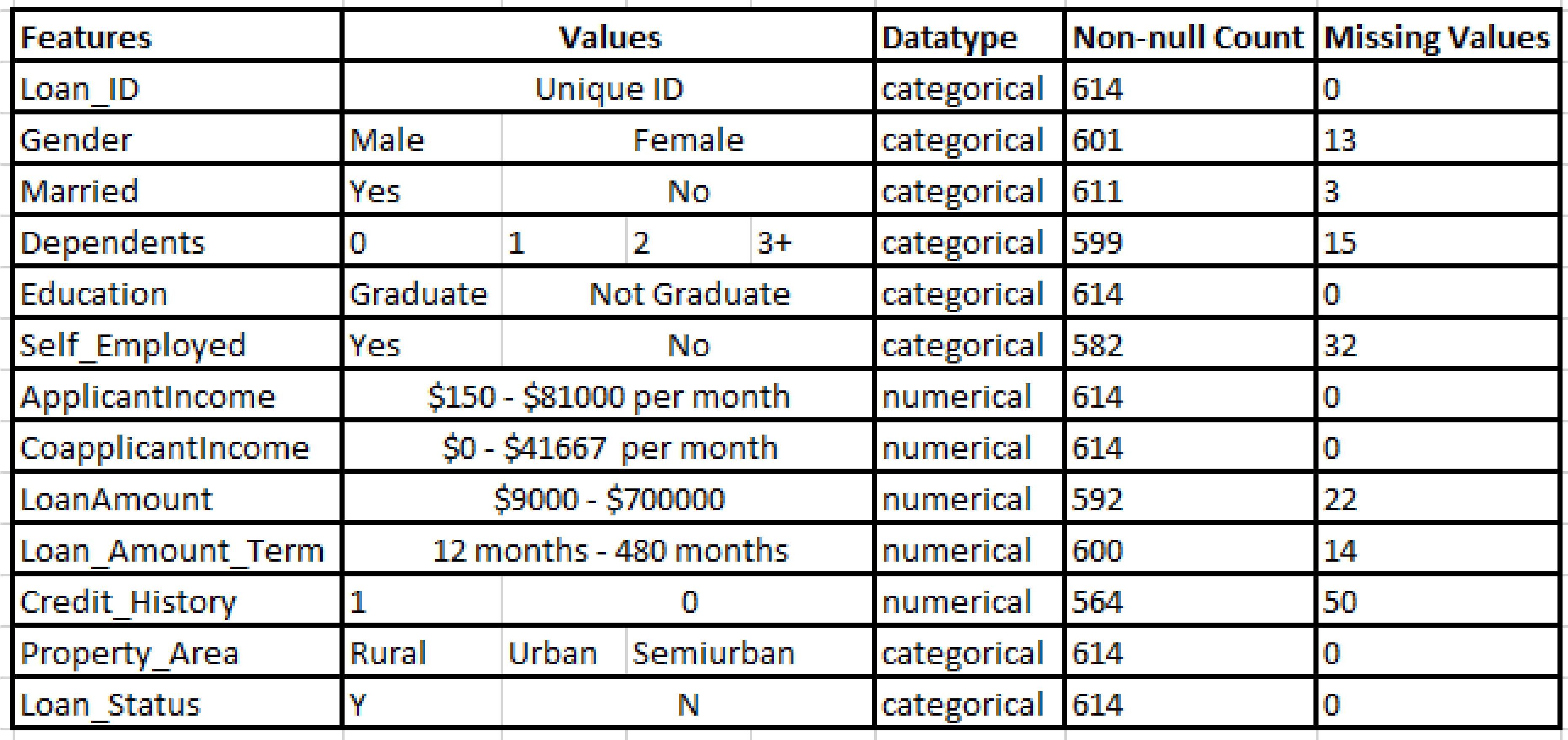
Dataset

The Loan Eligible dataset can be downloaded from Kaggle

It’s a short dataset with 13 columns and 614 rows

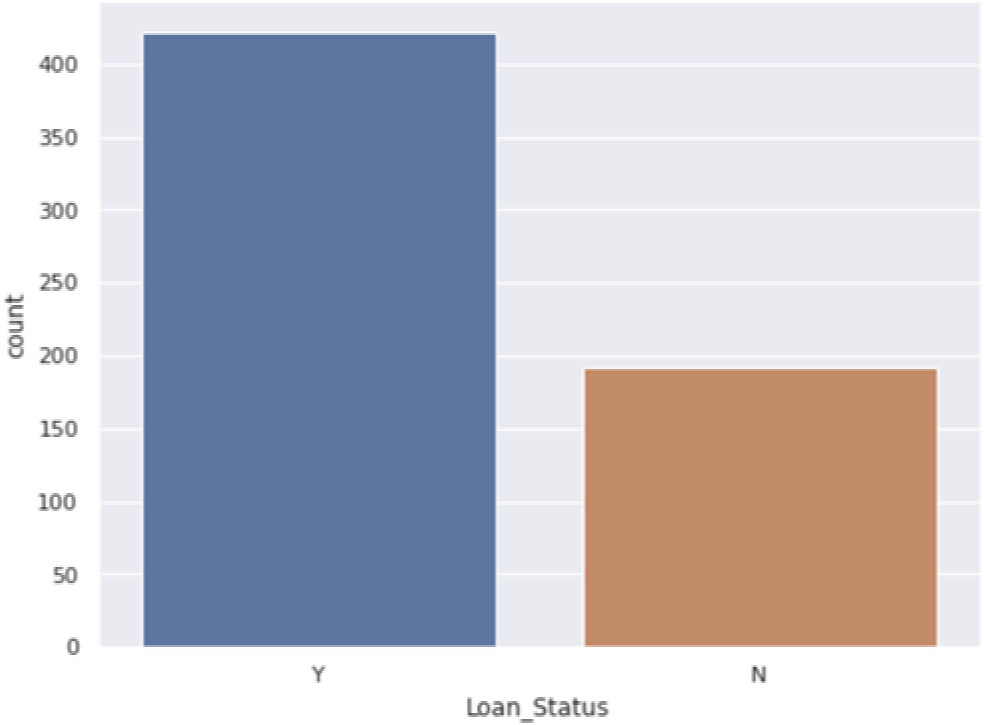
This dataset has some missing values and some of the features have a wide range of values, so we’ll need to preprocess it.

Dataset Features



Features	Values				Datatype	Non-null Count	Missing Values
Loan_ID	Unique ID				categorical	614	0
Gender	Male	Female			categorical	601	13
Married	Yes	No			categorical	611	3
Dependents	0	1	2	3+	categorical	599	15
Education	Graduate	Not Graduate			categorical	614	0
Self_Employed	Yes	No			categorical	582	32
ApplicantIncome	$150 - $81000 per month				numerical	614	0
CoapplicantIncome	$0 - $41667 per month				numerical	614	0
LoanAmount	$9000 - $700000				numerical	592	22
Loan_Amount_Term	12 months - 480 months				numerical	600	14
Credit_History	1	0			numerical	564	50
Property_Area	Rural	Urban	Semiurban		categorical	614	0
Loan_Status	Y	N			categorical	614	0

Dataset Visualization



Loan Status

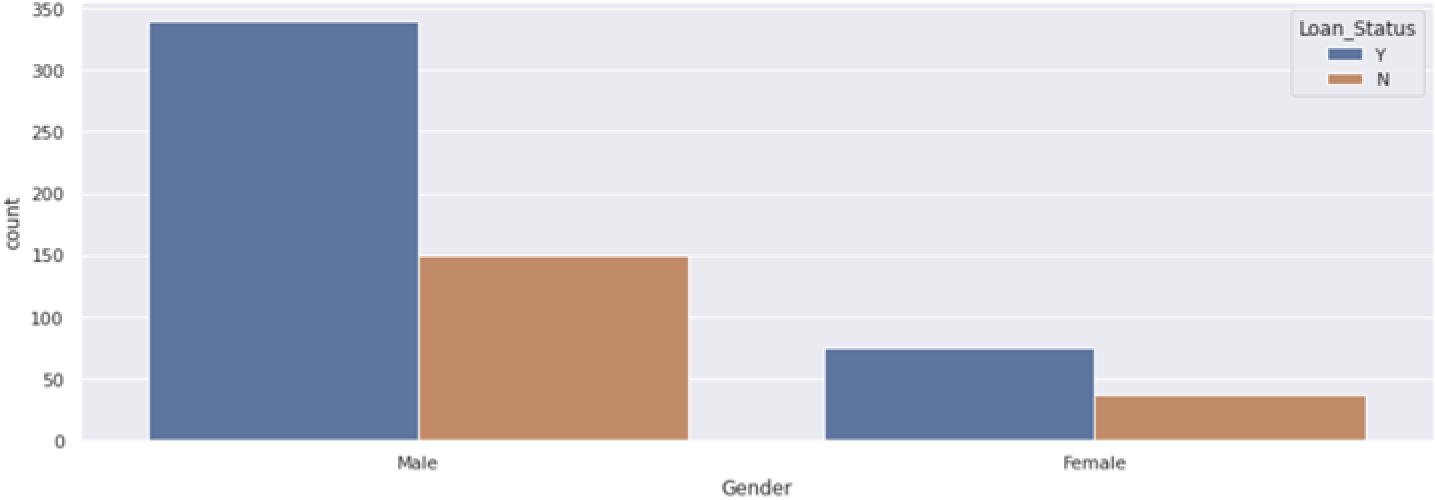
- Yes: 422 (69%)
- No: 192 (31%)

Dataset Visualization

Gender

Male: 489 (Y 69%, N 31%)

Female: 112 (Y 67%, N

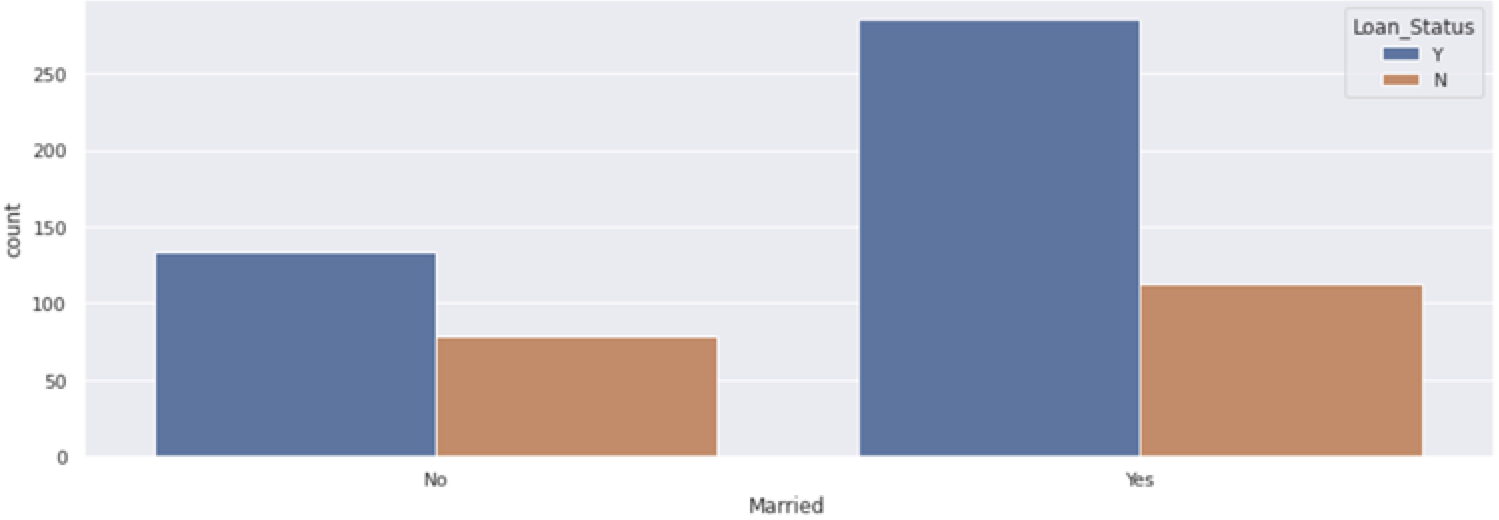


Dataset Visualization

Married

No: 213 (Y 63%, N 37%)

Yes: 398 (Y 72%, N 28%)

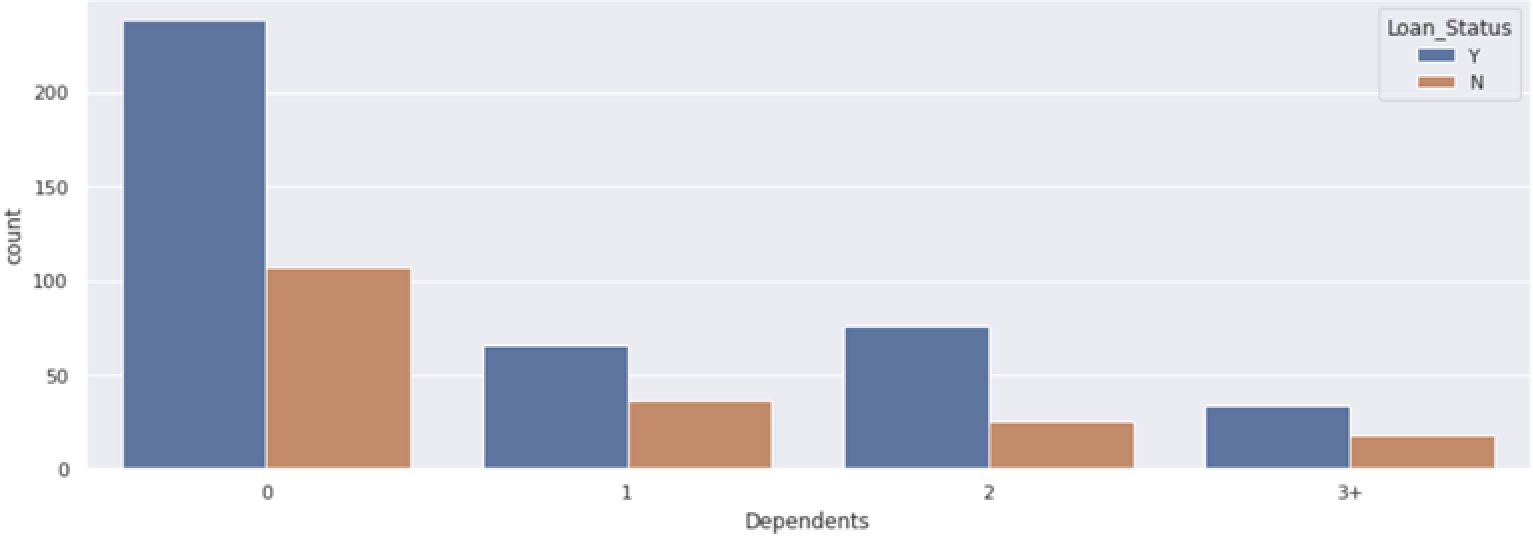


Dataset Visualization

Dependents

0: 345 (Y 69%, N 31%) 1: 102 (Y 65%, N 35%)

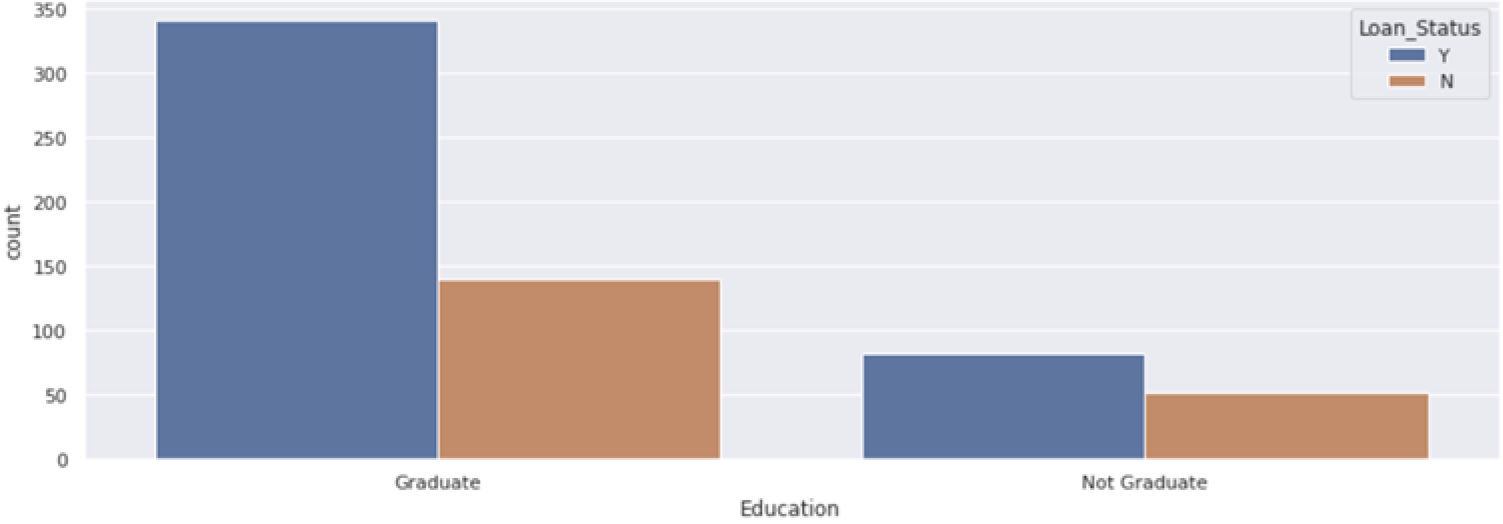
2: 101 (Y 76%, N 24%) 3+: 51 (Y 64%, N 36%)



Dataset Visualization

Education

Graduate: 480 (Y 71%, N 29%)   Not Graduate: 134 (Y 61%, N 39%)

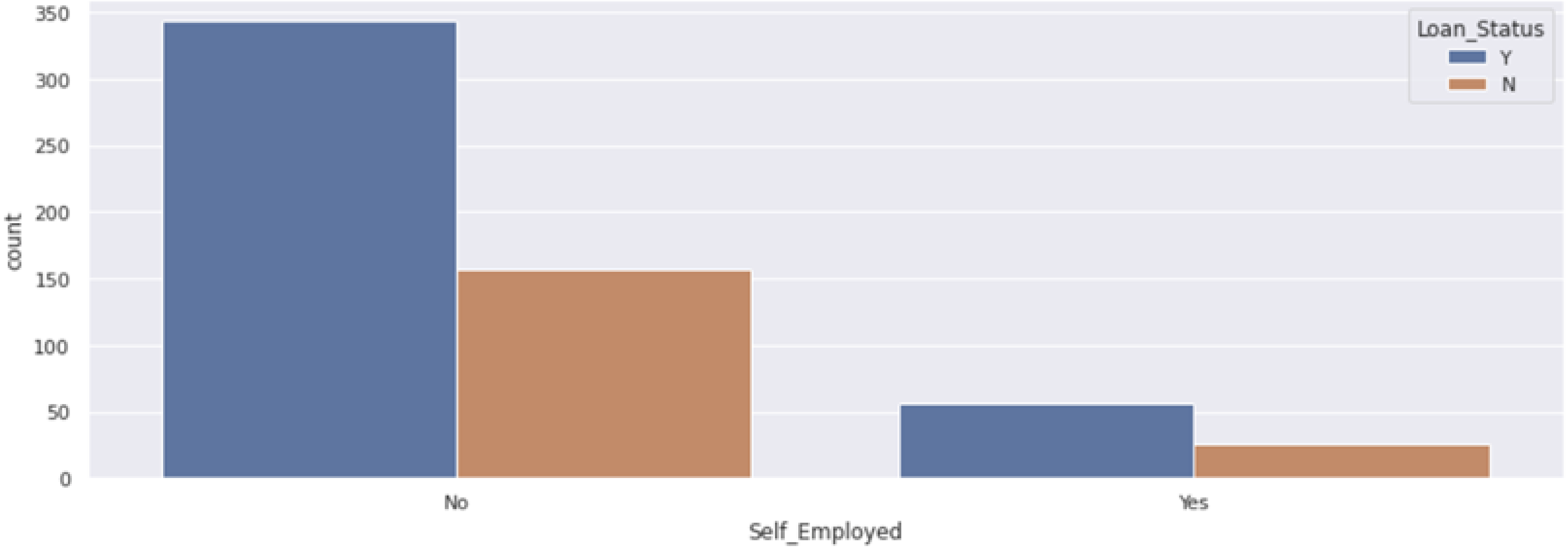


Dataset Visualization

Self-Employed

No: 500 (Y 69%, N 21%)

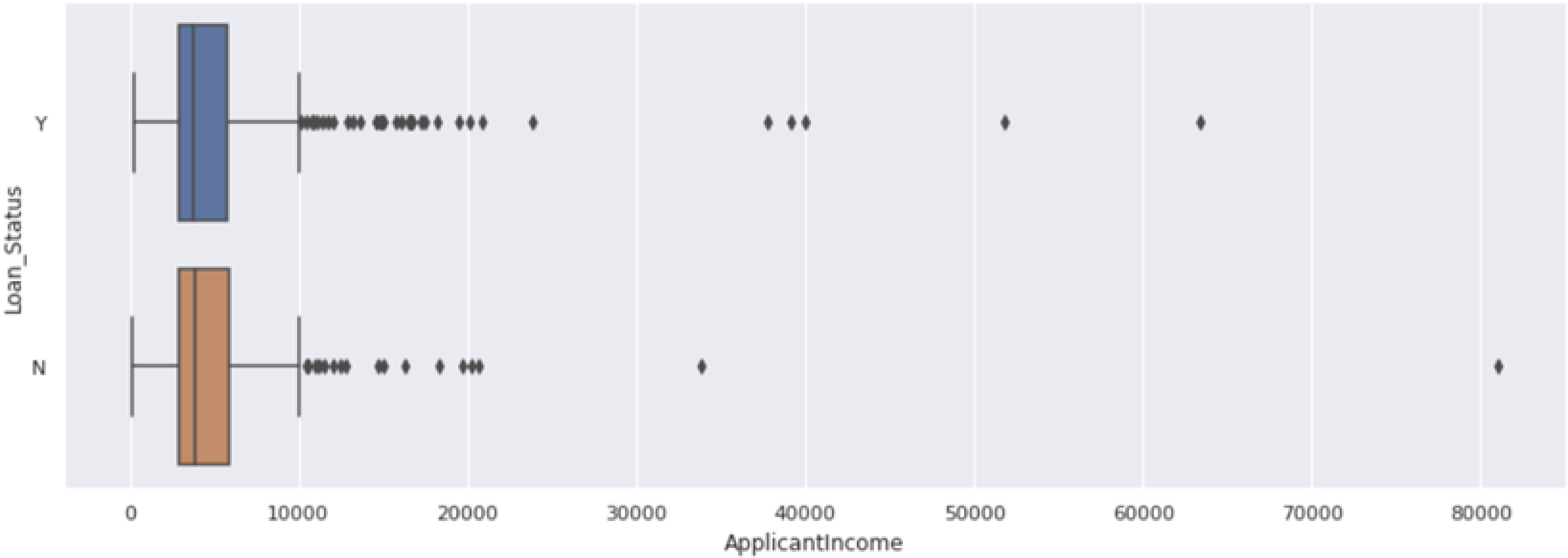
Yes: 82 (Y 68%, N 22%



Dataset Visualization

Applicant Income

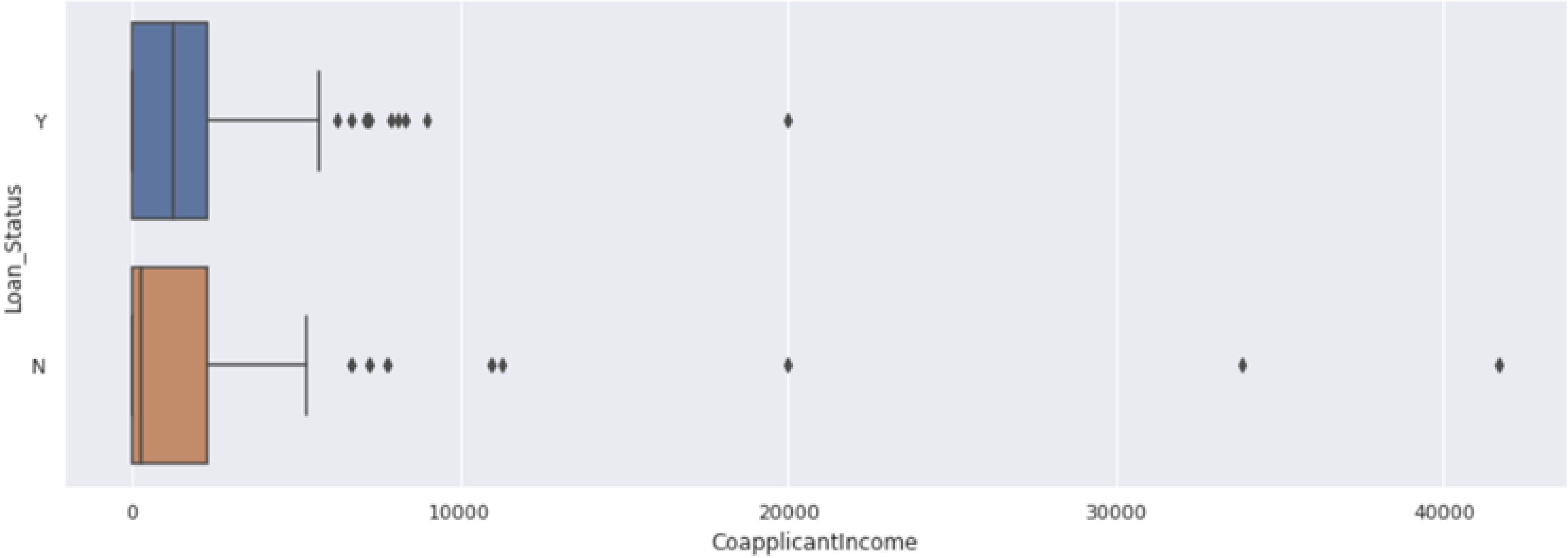
Min: 150 Max: 81000 25%: 2877 50%: 3812 75%: 5795 Mean: 5403



Dataset Visualization

Co-Applicant Income

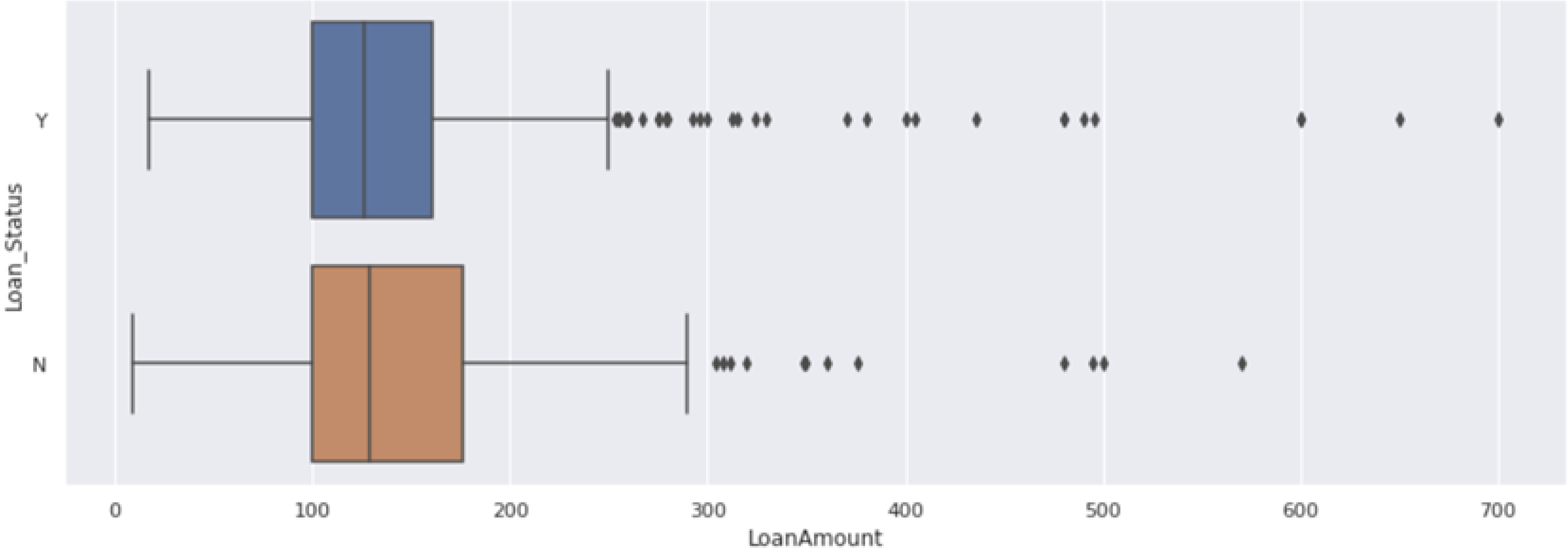
Min: 0 Max: 41667 25%: 0 50%: 1188 75%: 2297 Mean: 1621



Dataset Visualization

Loan Amount

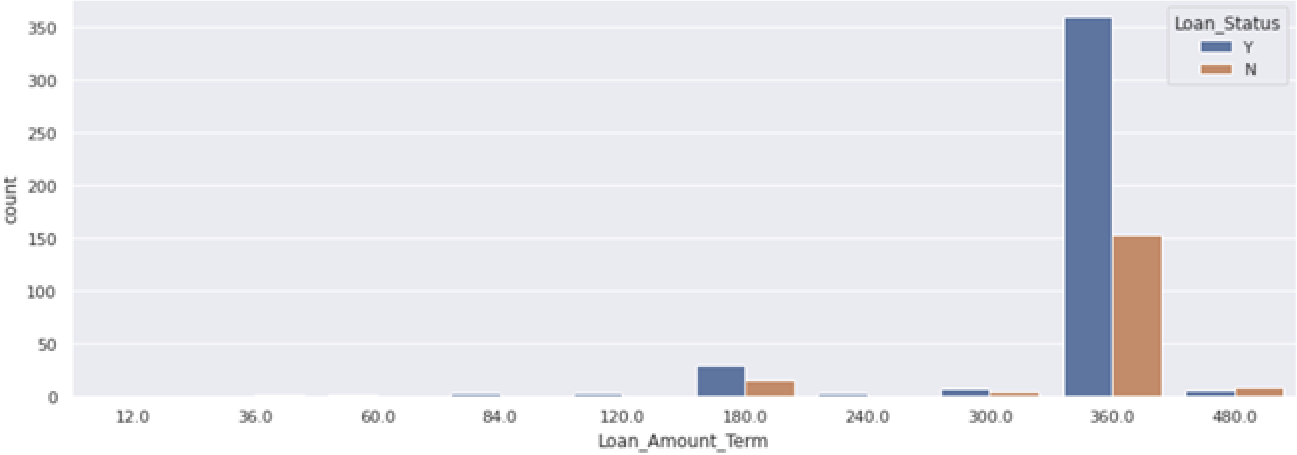
Min: 9k Max: 700k 25%: 100k 50%: 128k 75%: 168k Mean: 146k



Dataset Visualization

Loan Amount Term

0-7 years: 9 (Y 66%, N 34%) 10 years: 3 (Y 100%) 15 years: 44 (Y 66%, N 34%)
20 years: 4 (Y 75%, N 25%) 25 years: 13 (Y 62%, N 38%)
30 years: 512 (Y 70%, N 30%) 40 years: 15 (Y 40%, N 60%)

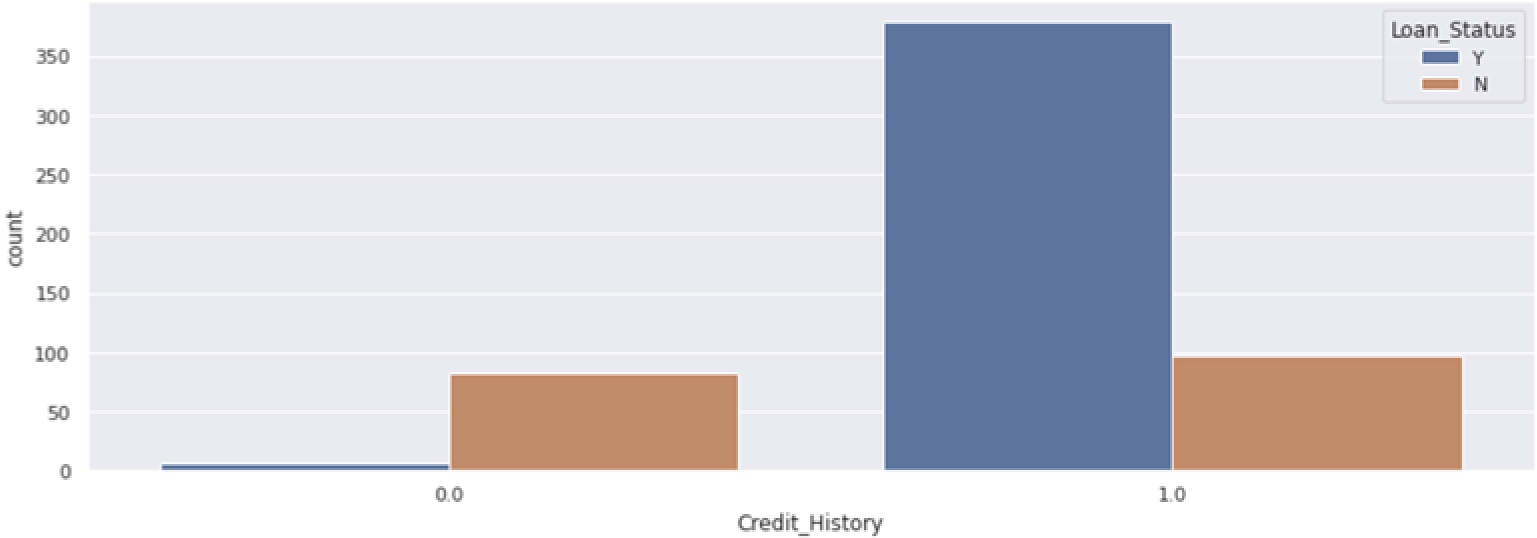


Dataset Visualization

Credit History

0: 89 (Y 8% , N 92%)

1: 475 (Y 80% , N 20%)



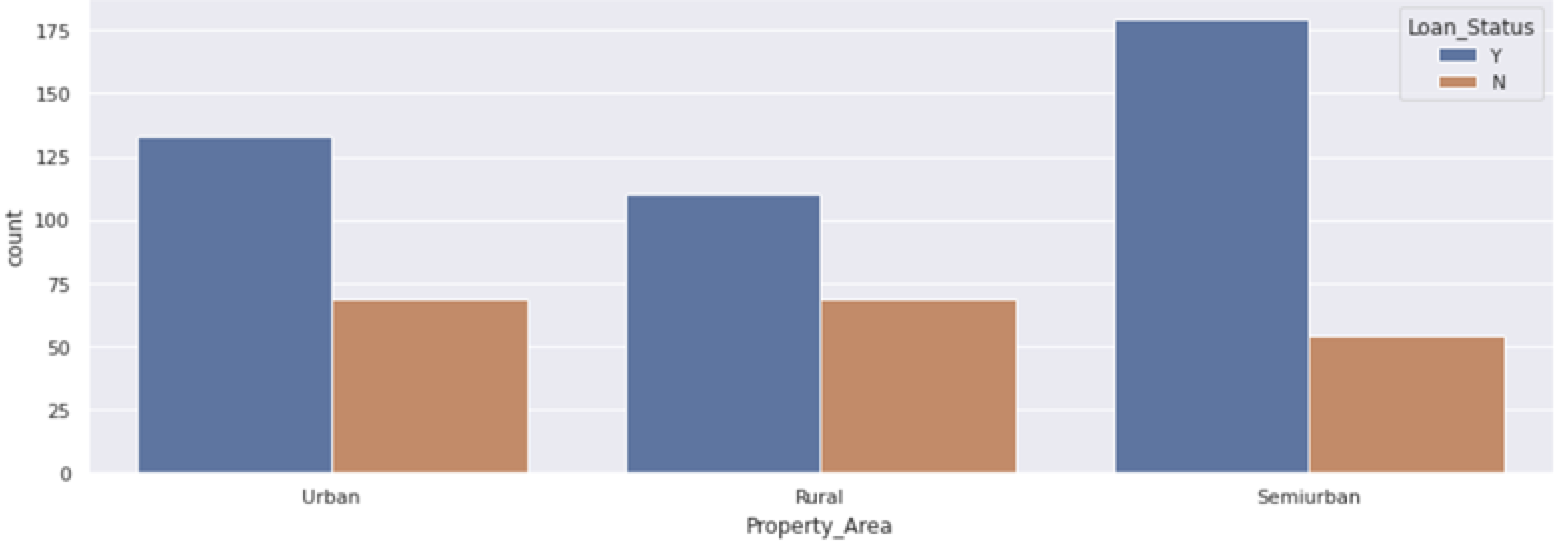
Dataset Visualization

Property Area

Urban: 202 (Y 66% , N 34%)

Rural: 179 (Y 61% , N 39%)

Semi-Urban: 233 (Y 77% , N 23%)



Preprocessing the Dataset

Fill up the missing values

- Mode for categorical features
- Mean or median for numerical features
- Zero for dependents: dependents can be categorized as missing not at random (if a person "forgot" to fill up the number of dependents, most likely they don't have dependents)
- Zero for credit history: credit history can be categorized as missing not at random (if a person "forgot" to fill up the credit history, most likely they don't have history)

Preprocessing the Dataset

Fill up the missing values

- Gender 13 → mode
- Married 3 → mode
- Dependents 15 → 0
- Self_Employed 32 → mode
- LoanAmount 22 → mean
- Loan_Amount_Term 14 → median
- Credit_History 50 → 0

Preprocessing the Dataset

Upsampling the dataset

Imbalanced Initial Dataset -

- 422 labels for Y (69%)
- 192 labels for N (31%)

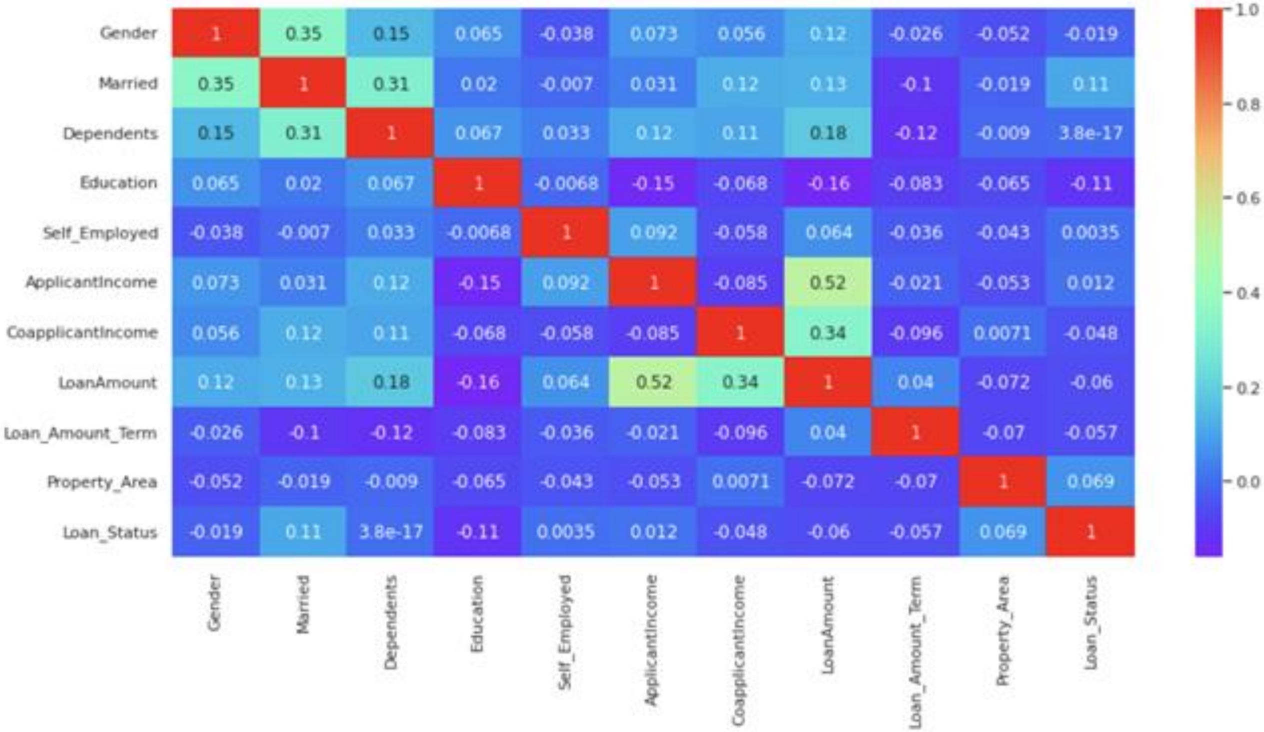
Upsampled Dataset -

- 422 labels for Y
- 422 labels for N

Total Count of Upsampled dataset = 844

Preprocessing the Dataset

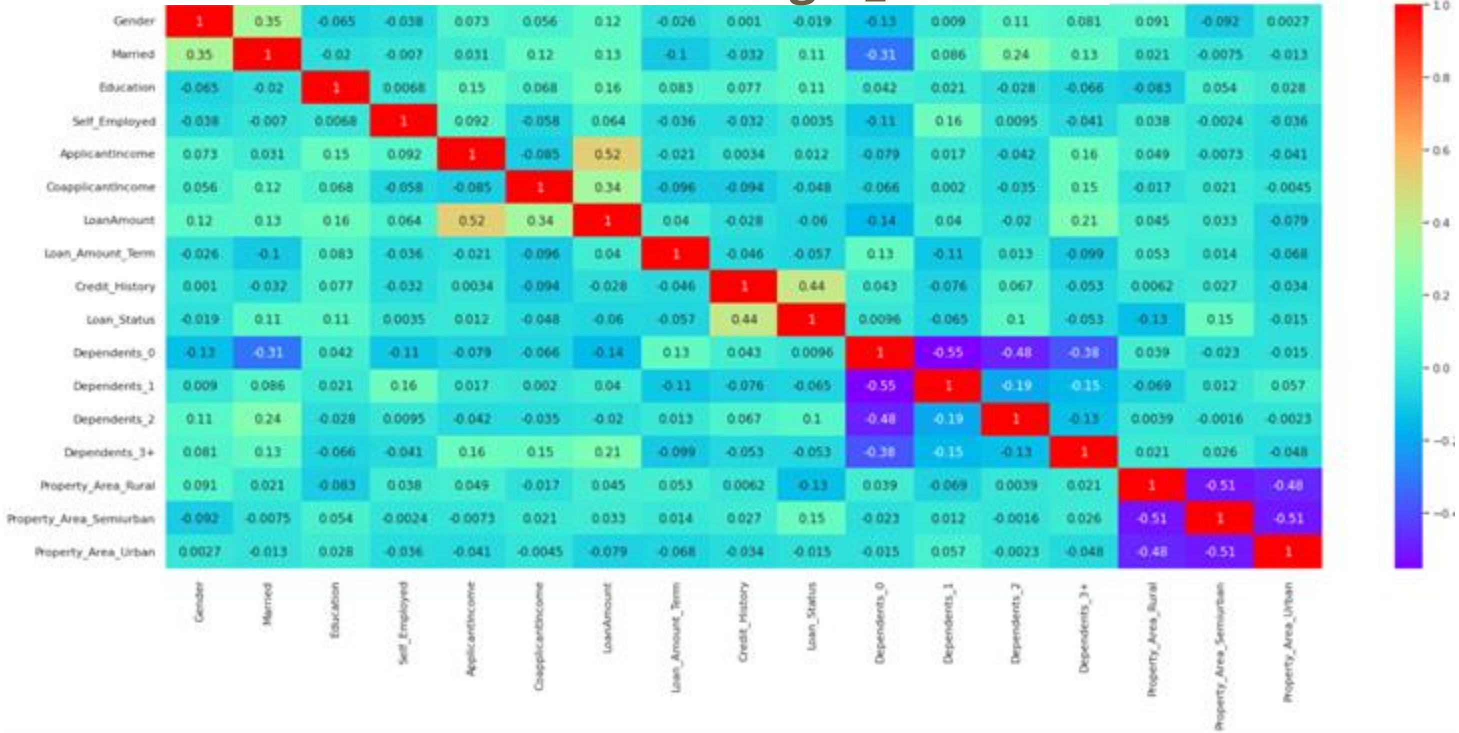
Categorical values into numerical with LabelEncoder



	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Property_Area	Loan_Status
Gender	1	0.35	0.15	0.065	-0.038	0.073	0.056	0.12	-0.026	-0.052	-0.019
Married	0.35	1	0.31	0.02	-0.007	0.031	0.12	0.13	-0.1	-0.019	0.11
Dependents	0.15	0.31	1	0.067	0.033	0.12	0.11	0.18	-0.12	-0.009	3.8e-17
Education	0.065	0.02	0.067	1	-0.0068	-0.15	-0.068	-0.16	-0.083	-0.065	-0.11
Self_Employed	-0.038	-0.007	0.033	-0.0068	1	0.092	-0.058	0.064	-0.036	-0.043	0.0035
ApplicantIncome	0.073	0.031	0.12	-0.15	0.092	1	-0.085	0.52	-0.021	-0.053	0.012
CoapplicantIncome	0.056	0.12	0.11	-0.068	-0.058	-0.085	1	0.34	-0.096	0.0071	-0.048
LoanAmount	0.12	0.13	0.18	-0.16	0.064	0.52	0.34	1	0.04	-0.072	-0.06
Loan_Amount_Term	-0.026	-0.1	-0.12	-0.083	-0.036	-0.021	-0.096	0.04	1	-0.07	-0.057
Property_Area	-0.052	-0.019	-0.009	-0.065	-0.043	-0.053	0.0071	-0.072	-0.07	1	0.069
Loan_Status	-0.019	0.11	3.8e-17	-0.11	0.0035	0.012	-0.048	-0.06	-0.057	0.069	1

Preprocessing the Dataset

Categorical values into numerical with get_dummies



	Gender	Married	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_Status	Dependents_0	Dependents_1	Dependents_2	Dependents_3+	Property_Area_Rural	Property_Area_Semiurban	Property_Area_Urban
Gender	1	0.35	-0.065	-0.038	0.073	0.056	0.12	-0.026	0.001	-0.019	-0.13	0.009	0.11	0.081	0.091	-0.092	0.0027
Married	0.35	1	-0.02	-0.007	0.031	0.12	0.13	-0.1	-0.032	0.11	-0.31	0.086	0.24	0.13	0.021	-0.0075	-0.013
Education	-0.065	-0.02	1	0.0068	0.15	0.068	0.16	0.083	0.077	0.11	0.042	0.021	-0.028	-0.066	-0.083	0.054	0.028
Self_Employed	-0.038	-0.007	0.0068	1	0.092	-0.058	0.064	-0.036	-0.032	0.0035	-0.11	0.16	0.0095	-0.041	0.038	-0.0024	-0.036
ApplicantIncome	0.073	0.031	0.15	0.092	1	-0.085	0.52	-0.021	0.0034	0.012	-0.079	0.017	-0.042	0.16	0.049	-0.0073	-0.041
CoapplicantIncome	0.056	0.12	0.068	-0.058	-0.085	1	0.34	-0.096	-0.094	-0.048	-0.066	0.002	-0.035	0.15	-0.017	0.021	-0.0045
LoanAmount	0.12	0.13	0.16	0.064	0.52	0.34	1	0.04	-0.028	-0.06	-0.14	0.04	-0.02	0.21	0.045	0.033	-0.079
Loan_Amount_Term	-0.026	-0.1	0.083	-0.036	-0.021	-0.096	0.04	1	-0.046	-0.057	0.13	-0.11	0.013	-0.099	0.053	0.014	-0.068
Credit_History	0.001	-0.032	0.077	-0.032	0.0034	-0.094	-0.028	-0.046	1	0.44	0.043	-0.076	0.067	-0.053	0.0062	0.027	-0.034
Loan_Status	-0.019	0.11	0.11	0.0035	0.012	-0.048	-0.06	-0.057	0.44	1	0.0096	-0.065	0.1	-0.053	-0.13	0.15	-0.015
Dependents_0	-0.13	-0.31	0.042	-0.11	-0.079	-0.066	-0.14	0.13	0.043	0.0096	1	-0.55	-0.48	-0.38	0.039	-0.023	-0.015
Dependents_1	0.009	0.086	0.021	0.16	0.017	0.002	0.04	-0.11	-0.076	-0.065	-0.55	1	-0.19	-0.15	-0.069	0.012	0.057
Dependents_2	0.11	0.24	-0.028	0.0095	-0.042	-0.035	-0.02	0.013	0.067	0.1	-0.48	-0.19	1	-0.13	0.0039	-0.0016	-0.0023
Dependents_3+	0.081	0.13	-0.066	-0.041	0.16	0.15	0.21	-0.099	-0.053	-0.053	-0.38	-0.15	-0.13	1	0.021	0.026	-0.048
Property_Area_Rural	0.091	0.021	-0.083	0.038	0.049	-0.017	0.045	0.053	0.0062	-0.13	0.039	-0.069	0.0039	0.021	1	-0.51	-0.48
Property_Area_Semiurban	-0.092	-0.0075	0.054	-0.0024	-0.0073	0.021	0.033	0.014	0.027	0.15	-0.023	0.012	-0.0016	0.026	-0.51	1	-0.51
Property_Area_Urban	0.0027	-0.013	0.028	-0.036	-0.041	-0.0045	-0.079	-0.068	-0.034	-0.015	-0.015	0.057	-0.0023	-0.048	-0.48	-0.51	1

Splitting the Dataset

Training -

- 70% of dataset
- 590 rows

Test -

- 30% of dataset
- 254 rows

No Validation dataset, since data available is less.

Use of k- fold cross validation to tune hyperparameters

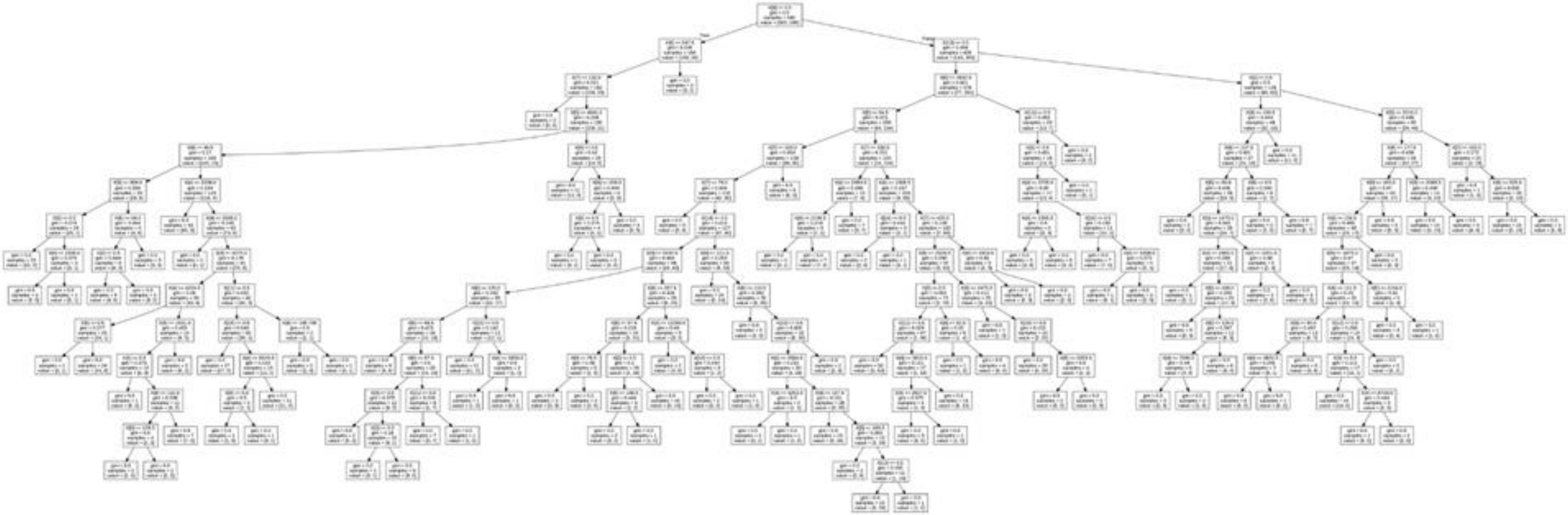
Methods

1. **Decision Tree Classifier**
    1. simple and easy to interpret
    2. trees can be visualized
2. **Random Forest Classifier**
    1. have much higher accuracy than the single decision tree
    2. doesn’t overfit the model, thus gives a good prediction on unseen datasets
    3. low bias and low variance
3. **Logistic Regression**
    1. efficient for linear dataset
    2. it can handle both dense and sparse input

Decision Tree Classifier

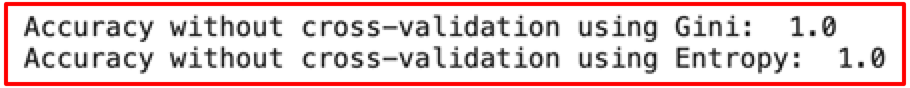
Default

Depth = 14



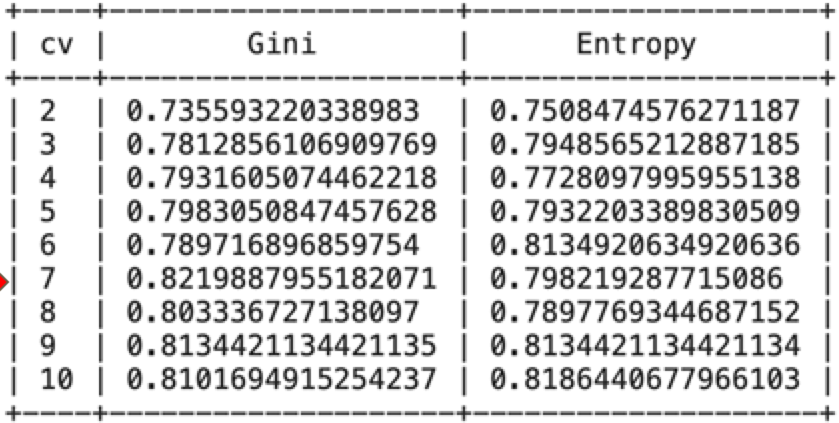
Decision Tree Classifier

Experiments: Gini vs Entropy, k-fold Cross Validation



```
Accuracy without cross-validation using Gini:   1.0
Accuracy without cross-validation using Entropy:   1.0
```

Overfitting??

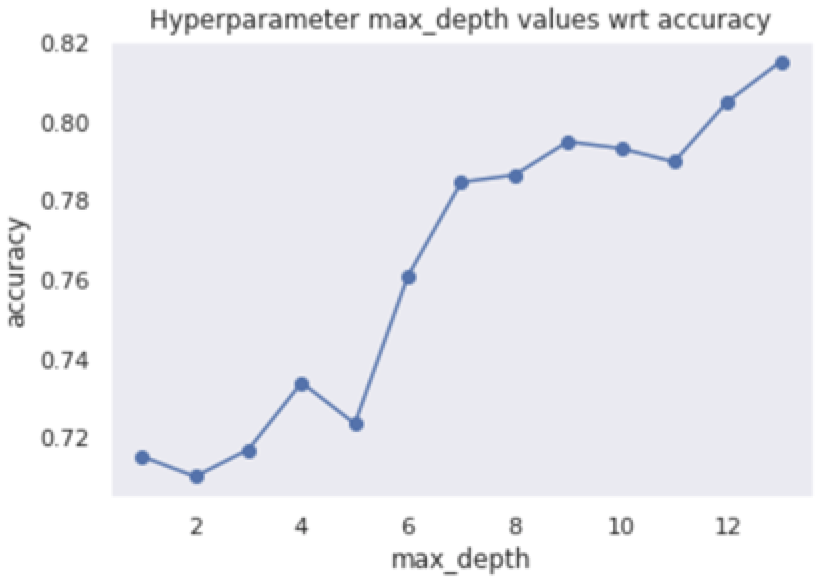


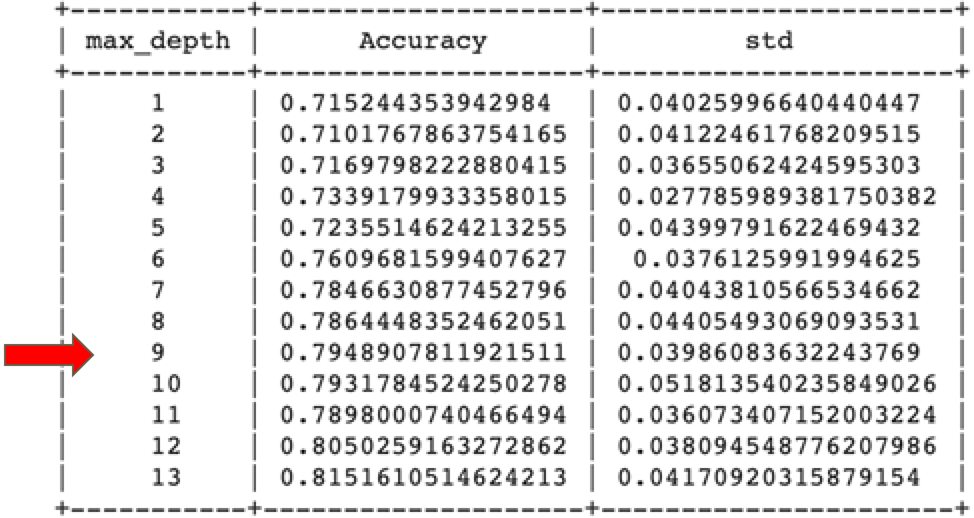
```
+----+--------------------+--------------------+
| cv |        Gini        |       Entropy      |
+----+--------------------+--------------------+
| 2  | 0.735593220338983  | 0.7508474576271187 |
| 3  | 0.7812856106909769 | 0.7948565212887185 |
| 4  | 0.7931605074462218 | 0.7728097995955138 |
| 5  | 0.7983050847457628 | 0.7932203389830509 |
| 6  | 0.789716896859754  | 0.8134920634920636 |
| 7  | 0.8219887955182071 | 0.798219287715086  |
| 8  | 0.803336727138097  | 0.7897769344687152 |
| 9  | 0.8134421134421135 | 0.8134421134421134 |
| 10 | 0.8101694915254237 | 0.8186440677966103 |
+----+--------------------+--------------------+
```

We select **k=7** and **Gini**

Decision Tree Classifier

Experiments: max_depth



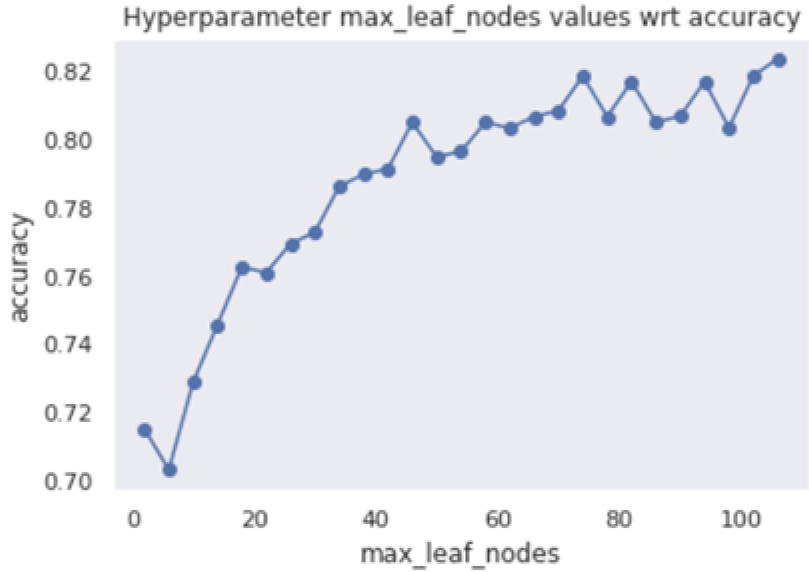


```
+------------+--------------------+----------------------+
| max_depth  |      Accuracy      |         std          |
+------------+--------------------+----------------------+
|     1      | 0.715244353942984  | 0.04025996640440447  |
|     2      | 0.7101767863754165 | 0.04122461768209515  |
|     3      | 0.7169798222880415 | 0.03655062424595303  |
|     4      | 0.7339179933358015 | 0.027785989381750382 |
|     5      | 0.7235514624213255 | 0.04399791622469432  |
|     6      | 0.7609681599407627 |  0.0376125991994625  |
|     7      | 0.7846630877452796 | 0.04043810566534662  |
|     8      | 0.7864448352462051 | 0.04405493069093531  |
|     9      | 0.7948907811921511 | 0.03986083632243769  |
|     10     | 0.7931784524250278 | 0.051813540235849026 |
|     11     | 0.7898000740466494 | 0.036073407152003224 |
|     12     | 0.8050259163272862 | 0.038094548776207986 |
|     13     | 0.8151610514624213 | 0.04170920315879154  |
+------------+--------------------+----------------------+
```

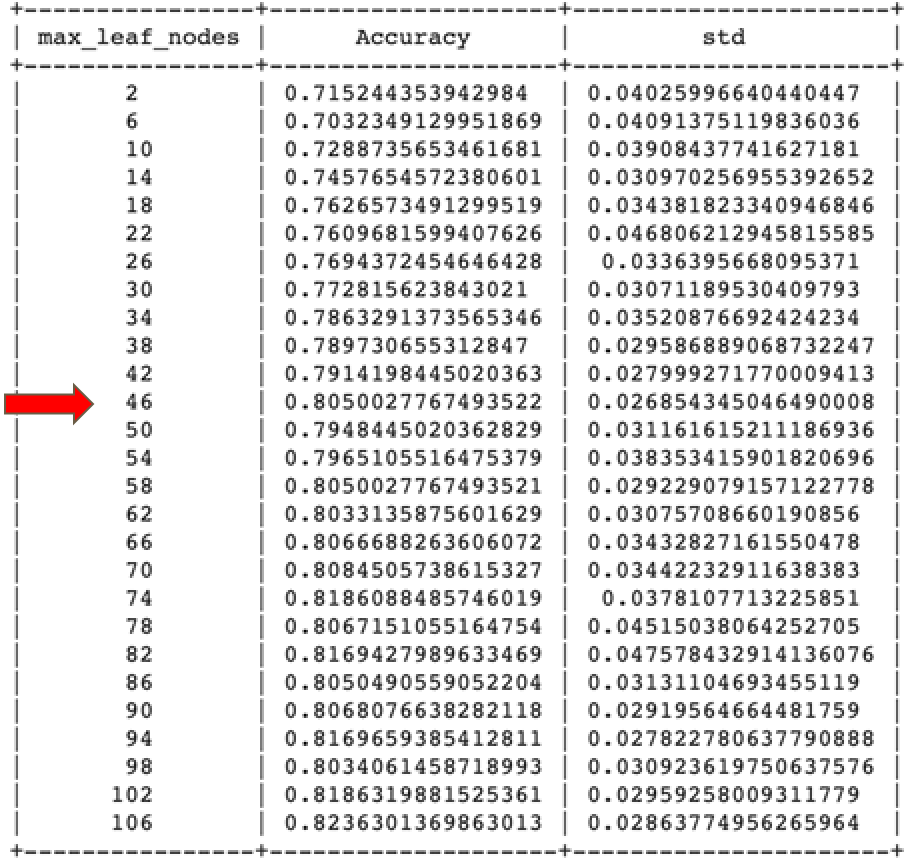
We select **max_depth=9**

Decision Tree Classifier

Experiments: max_leaf_nodes



We select **max_leaf_nodes=46**

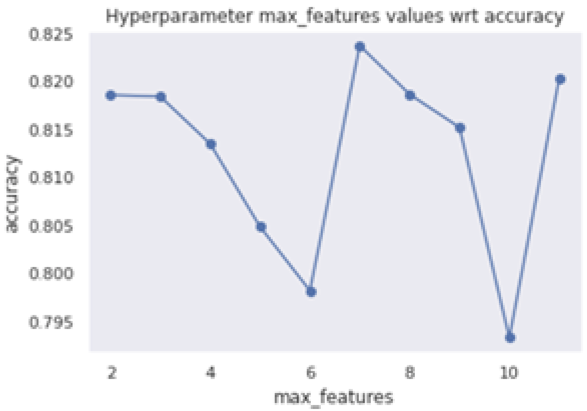


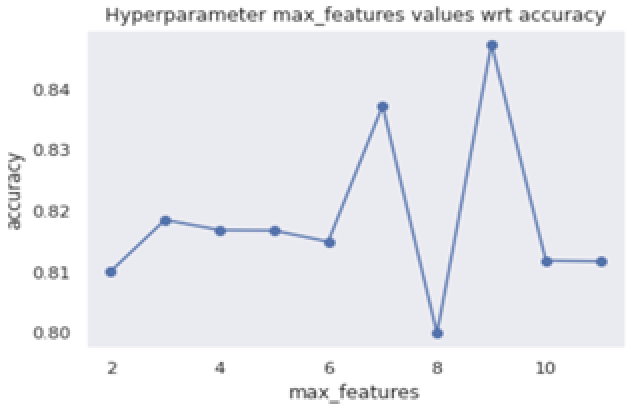
```
+----------------+--------------------+----------------------+
| max_leaf_nodes |      Accuracy      |         std          |
+----------------+--------------------+----------------------+
|       2        | 0.715244353942984  | 0.04025996640440447  |
|       6        | 0.7032349129951869 | 0.04091375119836036  |
|       10       | 0.7288735653461681 | 0.03908437741627181  |
|       14       | 0.7457654572380601 | 0.030970256955392652 |
|       18       | 0.7626573491299519 | 0.034381823340946846 |
|       22       | 0.7609681599407626 | 0.046806212945815585 |
|       26       | 0.7694372454646428 |  0.0336395668095371  |
|       30       | 0.772815623843021  | 0.03071189530409793  |
|       34       | 0.7863291373565346 | 0.03520876692424234  |
|       38       | 0.789730655312847  | 0.029586889068732247 |
|       42       | 0.7914198445020363 | 0.027999271770009413 |
|       46       | 0.8050027767493522 | 0.026854345046490008 |
|       50       | 0.7948445020362829 | 0.031161615211186936 |
|       54       | 0.7965105516475379 | 0.038353415901820696 |
|       58       | 0.8050027767493521 | 0.029222907915712278 |
|       62       | 0.8033135875601629 | 0.03075708660190856  |
|       66       | 0.8066688263606072 | 0.03432827161550478  |
|       70       | 0.8084505738615327 | 0.03442232911638383  |
|       74       | 0.8186088485746019 |  0.0378107713225851  |
|       78       | 0.8067151055164754 | 0.04515038064252705  |
|       82       | 0.8169427989633469 | 0.047578432914136076 |
|       86       | 0.8050490559052204 | 0.03131104693455119  |
|       90       | 0.8068076638282118 | 0.02919564664481759  |
|       94       | 0.8169659385412811 | 0.027822780637790888 |
|       98       | 0.8034061458718993 | 0.030923619750637576 |
|      102       | 0.8186319881525361 | 0.02959258009311779  |
|      106       | 0.8236301369863013 | 0.02863774956265964  |
+----------------+--------------------+----------------------+
```

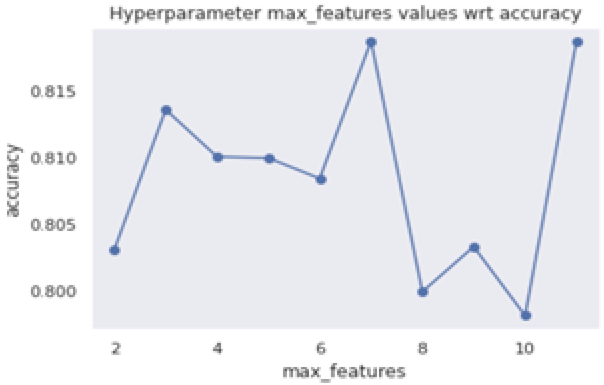
Decision Tree Classifier

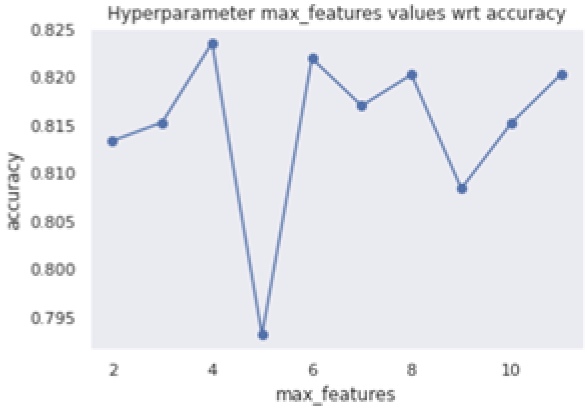
Experiments: max_features

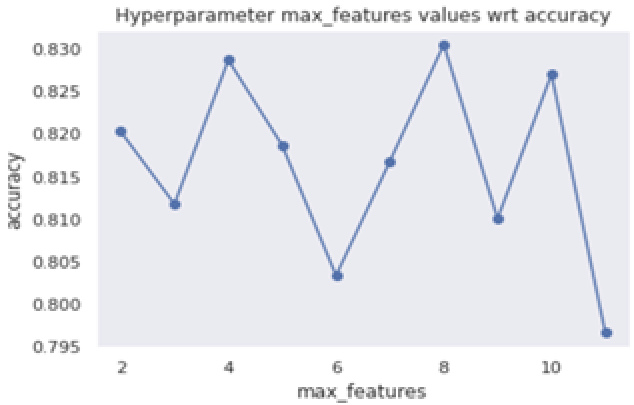
Random choices





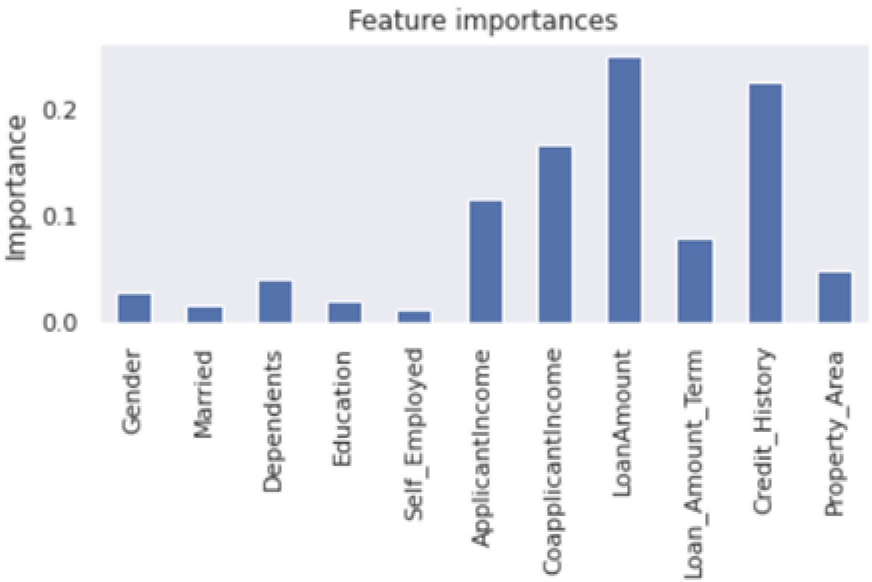


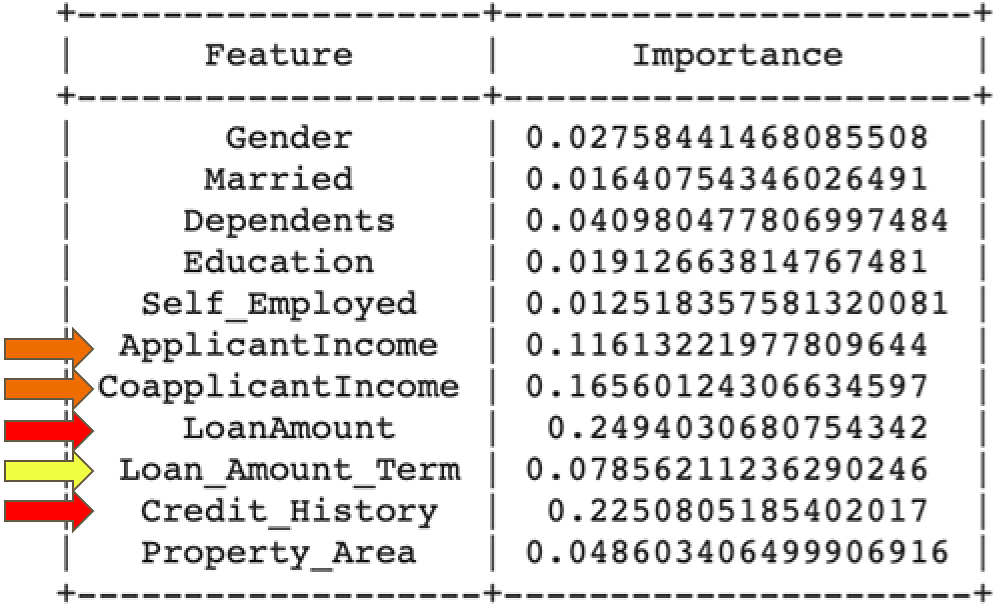




Decision Tree Classifier

Experiments: feature_importances_





```
+---------------------+----------------------+
|       Feature       |      Importance      |
+---------------------+----------------------+
|        Gender       | 0.02758441468085508  |
|       Married       | 0.01640754346026491  |
|      Dependents     | 0.040980477806997484 |
|      Education      | 0.01912663814767481  |
|    Self_Employed    | 0.012518357581320081 |
|   ApplicantIncome   | 0.11613221977809644  |
|  CoapplicantIncome  | 0.16560124306634597  |
|      LoanAmount     |  0.2494030680754342  |
|   Loan_Amount_Term  | 0.07856211236290246  |
|    Credit_History   |  0.2250805185402017  |
|    Property_Area    | 0.048603406499906916 |
+---------------------+----------------------+
```

Decision Tree Classifier

Experiments: feature_importances_

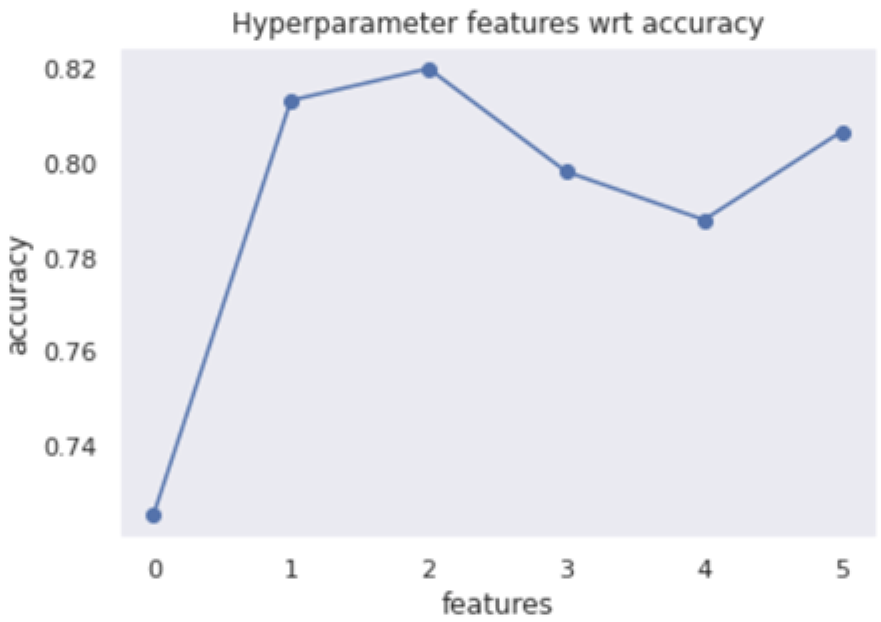
LoanAmount

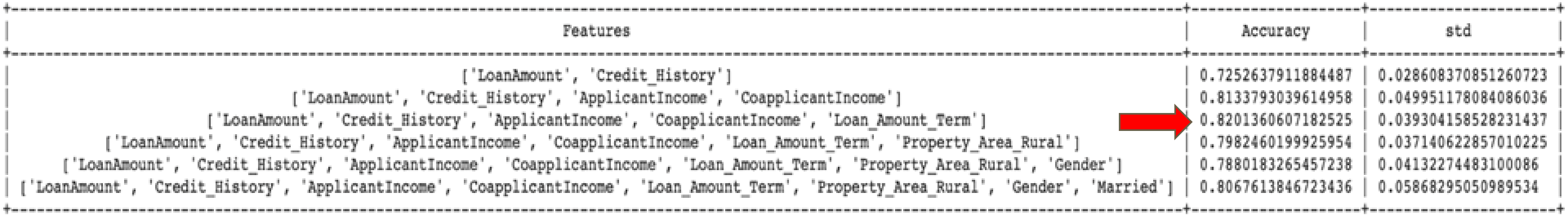
Credit_History

ApplicantIncome

CoapplicantIncome

Loan_Amount_Term

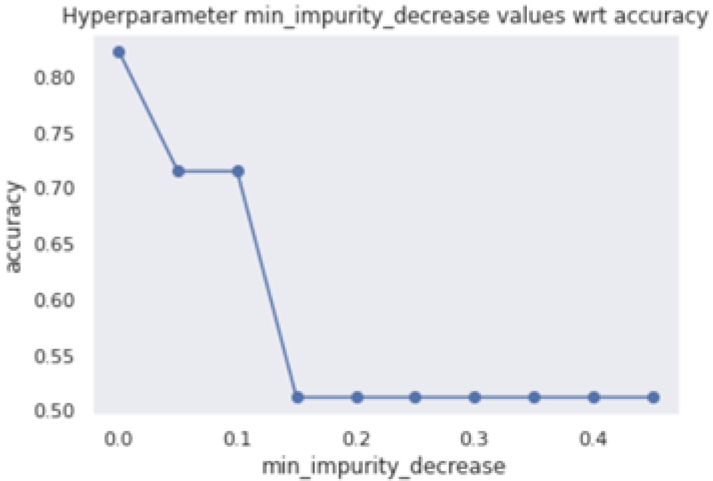


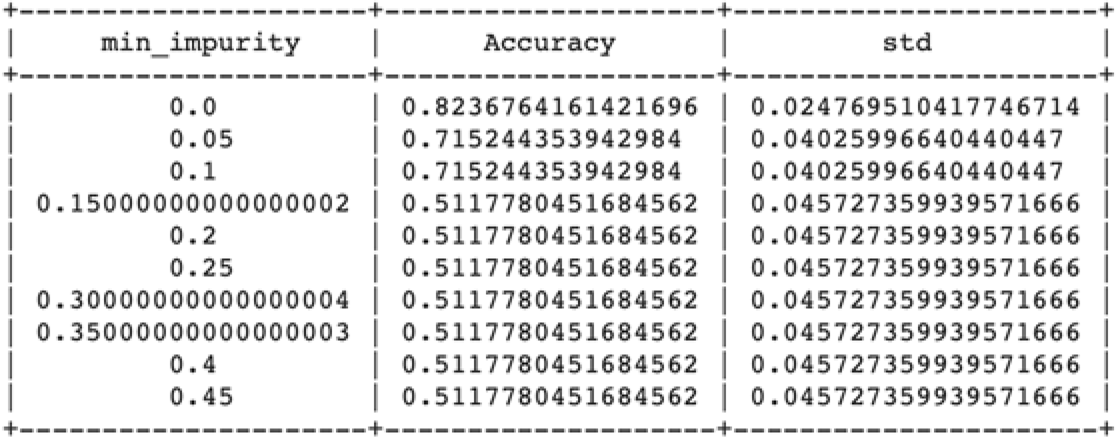


```
+------------------------------------------------------------------------------------------------------------------------------------------+--------------------+----------------------+
|                                                                 Features                                                                 |      Accuracy      |         std          |
+------------------------------------------------------------------------------------------------------------------------------------------+--------------------+----------------------+
|                                                     ['LoanAmount', 'Credit_History']                                                     | 0.7252637911884487 | 0.028608370851260723 |
|                                 ['LoanAmount', 'Credit_History', 'ApplicantIncome', 'CoapplicantIncome']                                 | 0.8133793039614958 | 0.049951178084086036 |
|                       ['LoanAmount', 'Credit_History', 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term']                       | 0.8201360607182525 | 0.039304158528231437 |
|           ['LoanAmount', 'Credit_History', 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term', 'Property_Area_Rural']            | 0.7982460199925954 | 0.037140622857010225 |
|      ['LoanAmount', 'Credit_History', 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term', 'Property_Area_Rural', 'Gender']       | 0.7880183265457238 | 0.04132274483100086  |
| ['LoanAmount', 'Credit_History', 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term', 'Property_Area_Rural', 'Gender', 'Married'] | 0.8067613846723436 | 0.05868295050989534  |
+------------------------------------------------------------------------------------------------------------------------------------------+--------------------+----------------------+
```

Decision Tree Classifier

Experiments: min_impurity



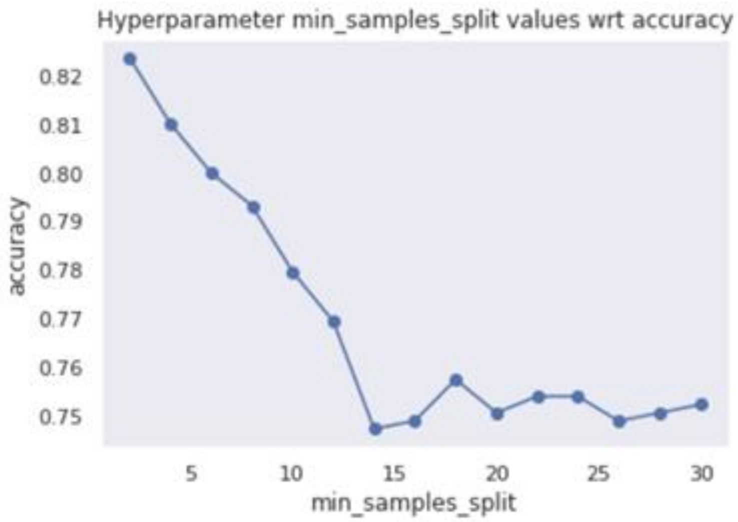


```
+---------------------+--------------------+----------------------+
|     min_impurity    |      Accuracy      |         std          |
+---------------------+--------------------+----------------------+
|         0.0         | 0.8236764161421696 | 0.024769510417746714 |
|         0.05        | 0.715244353942984  | 0.04025996640440447  |
|         0.1         | 0.715244353942984  | 0.04025996640440447  |
| 0.15000000000000002 | 0.5117780451684562 | 0.045727359939571666 |
|         0.2         | 0.5117780451684562 | 0.045727359939571666 |
|         0.25        | 0.5117780451684562 | 0.045727359939571666 |
| 0.30000000000000004 | 0.5117780451684562 | 0.045727359939571666 |
| 0.35000000000000003 | 0.5117780451684562 | 0.045727359939571666 |
|         0.4         | 0.5117780451684562 | 0.045727359939571666 |
|         0.45        | 0.5117780451684562 | 0.045727359939571666 |
+---------------------+--------------------+----------------------+
```

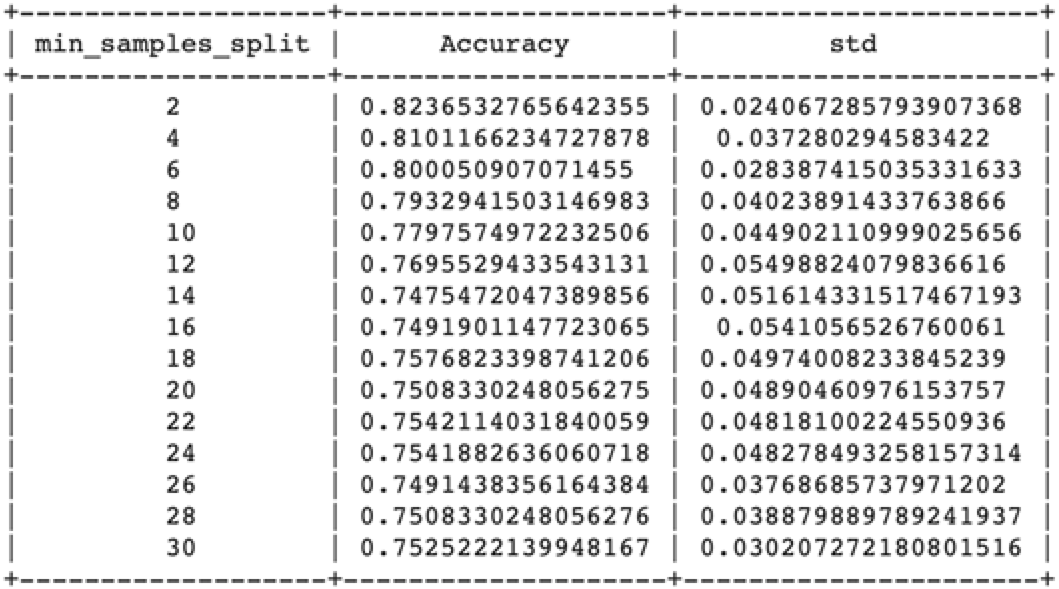
No improvement

Decision Tree Classifier

Experiments: min_smaples_split



No improvement

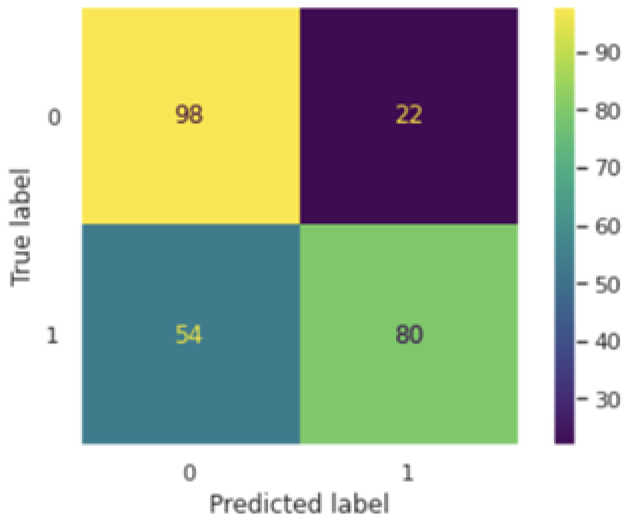


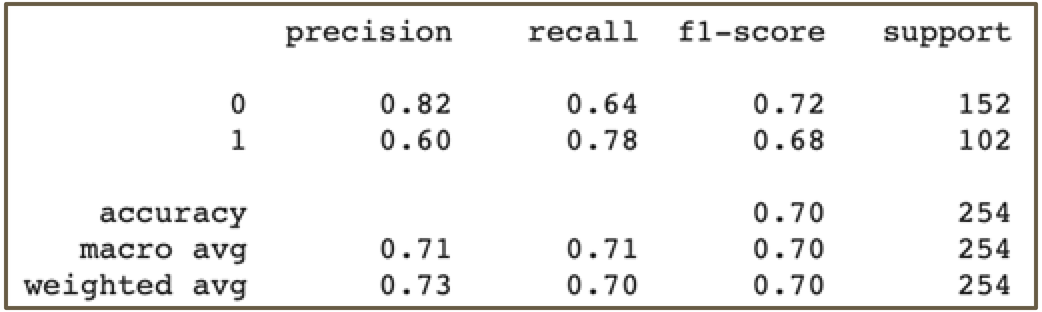
```
+-------------------+--------------------+----------------------+
| min_samples_split |      Accuracy      |         std          |
+-------------------+--------------------+----------------------+
|         2         | 0.8236532765642355 | 0.024067285793907368 |
|         4         | 0.8101166234727878 |  0.037280294583422   |
|         6         | 0.800050907071455  | 0.028387415035331633 |
|         8         | 0.7932941503146983 | 0.04023891433763866  |
|         10        | 0.7797574972232506 | 0.044902110999025656 |
|         12        | 0.7695529433543131 | 0.05498824079836616  |
|         14        | 0.7475472047389856 | 0.051614331517467193 |
|         16        | 0.7491901147723065 |  0.0541056526760061  |
|         18        | 0.7576823398741206 | 0.04974008233845239  |
|         20        | 0.7508330248056275 | 0.04890460976153757  |
|         22        | 0.7542114031840059 | 0.04818100224550936  |
|         24        | 0.7541882636060718 | 0.048278493258157314 |
|         26        | 0.7491438356164384 | 0.03768685737971202  |
|         28        | 0.7508330248056276 | 0.038879889789241937 |
|         30        | 0.7525222139948167 | 0.030207272180801516 |
+-------------------+--------------------+----------------------+
```

Decision Tree Classifier

Evaluation: criterion (Gini), cv (8-fold), max depth (9), max_leaf_nodes (46), features ['LoanAmount', 'Credit_History', 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term']

Confusion Matrix:

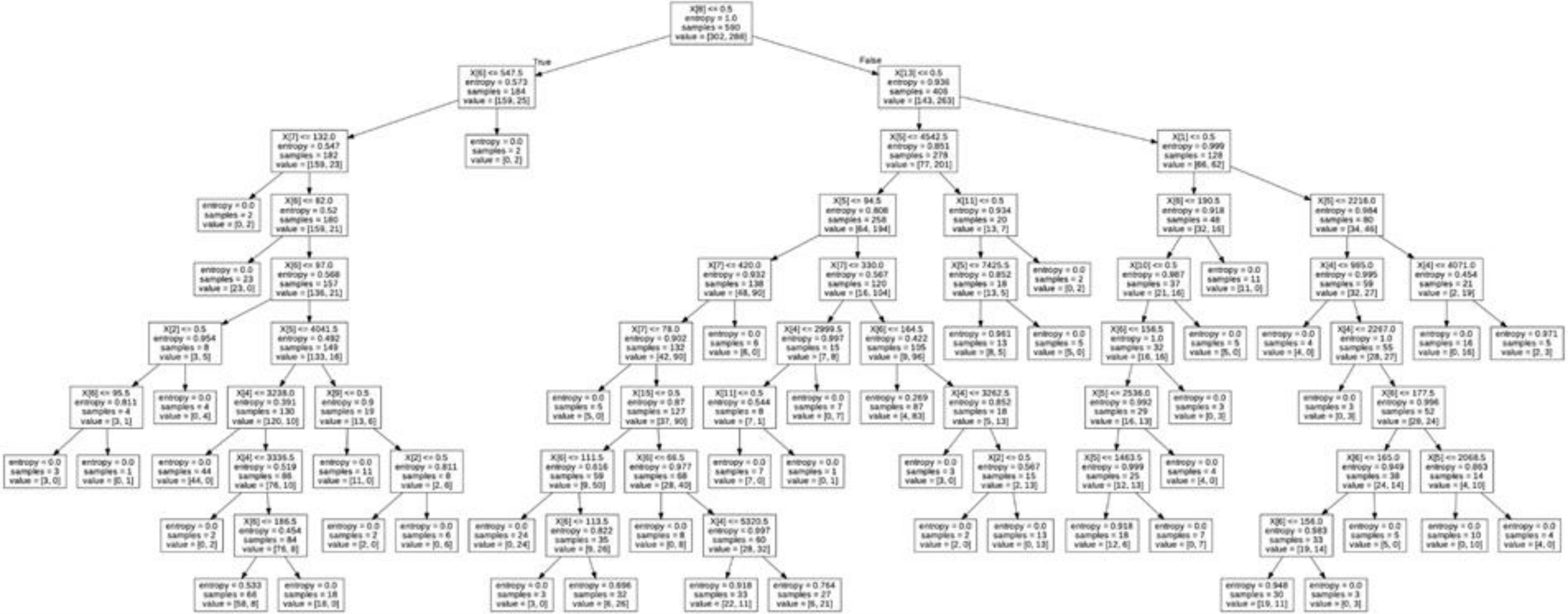




```
              precision    recall  f1-score   support

           0       0.82      0.64      0.72       152
           1       0.60      0.78      0.68       102

    accuracy                           0.70       254
   macro avg       0.71      0.71      0.70       254
weighted avg       0.73      0.70      0.70       254
```

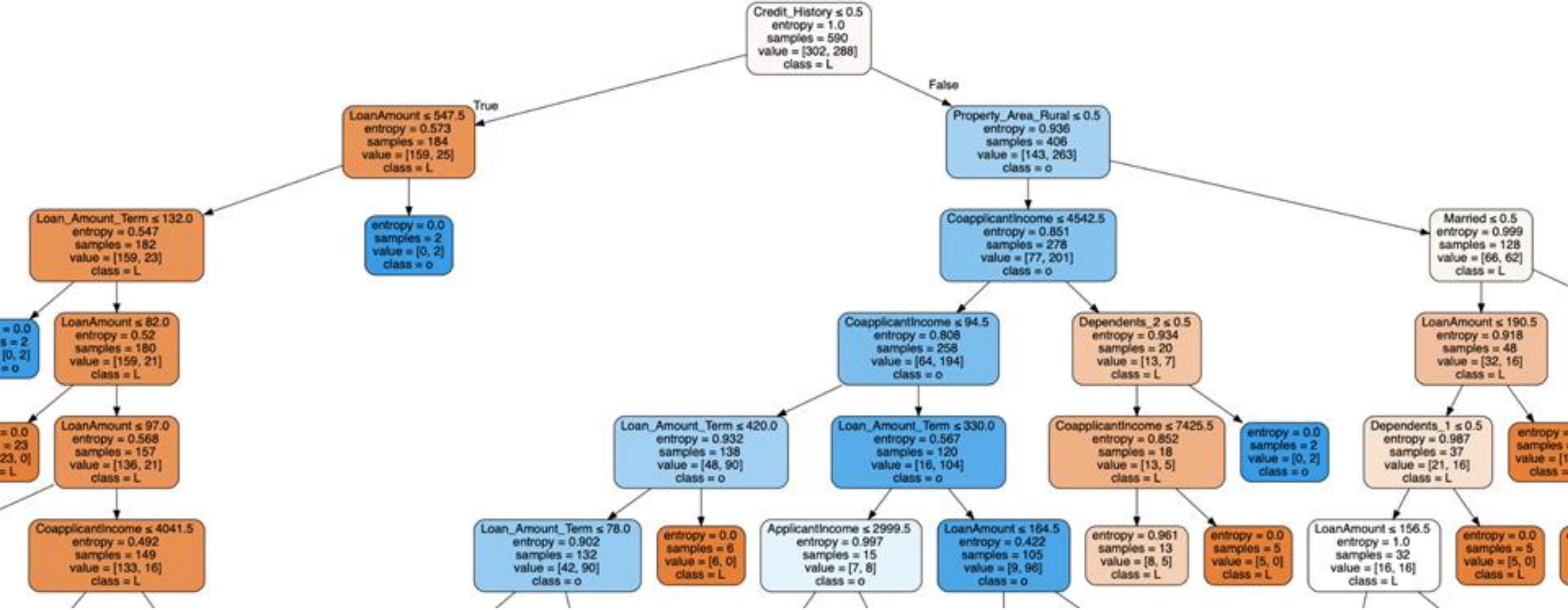
Decision Tree Classifier

Final Tree



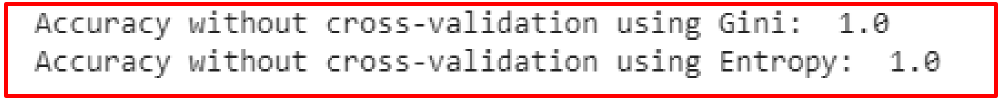
Decision Tree Classifier

Final Tree



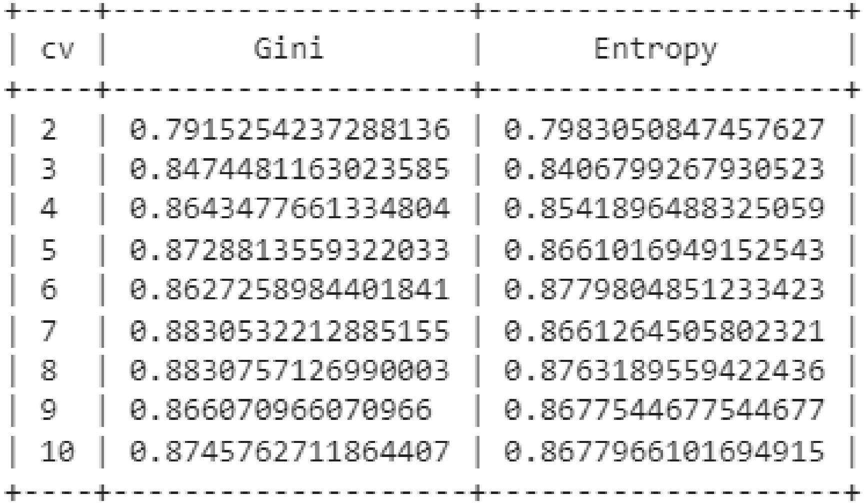
Random Forest Classifier

Experiment : Gini vs Entropy, k-fold Cross Validation



```
Accuracy without cross-validation using Gini:   1.0
Accuracy without cross-validation using Entropy:   1.0
```

Overfitting??

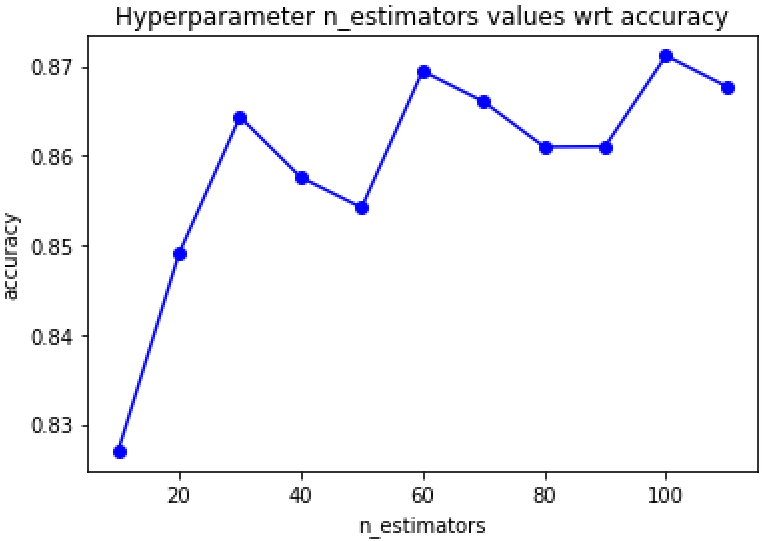


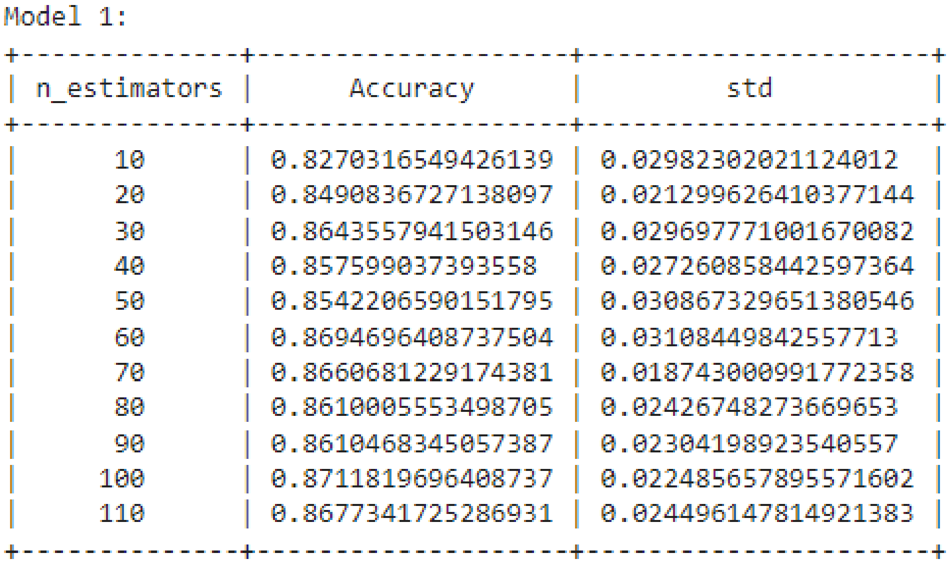
```
+----+--------------------+--------------------+
| cv |        Gini        |      Entropy       |
+----+--------------------+--------------------+
| 2  | 0.7915254237288136 | 0.7983050847457627 |
| 3  | 0.8474481163023585 | 0.8406799267930523 |
| 4  | 0.8643477661334804 | 0.8541896488325059 |
| 5  | 0.8728813559322033 | 0.8661016949152543 |
| 6  | 0.8627258984401841 | 0.8779804851233423 |
| 7  | 0.883053221288515  | 0.8661264505802321 |
| 8  | 0.8830757126990003 | 0.8763189559422436 |
| 9  | 0.866070966070966  | 0.8677544677544677 |
| 10 | 0.8745762711864407 | 0.8677966101694915 |
+----+--------------------+--------------------+
```

We select **k=8** and **Gini**

Random Forest Classifier

Experiment : tune n_estimators



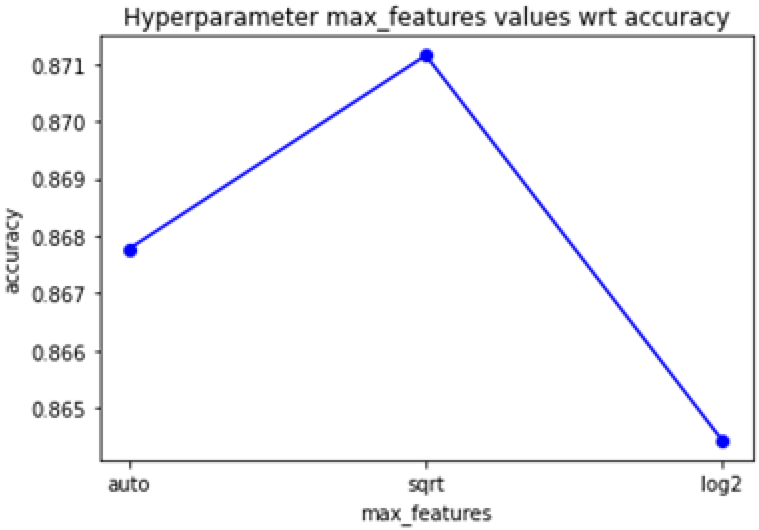


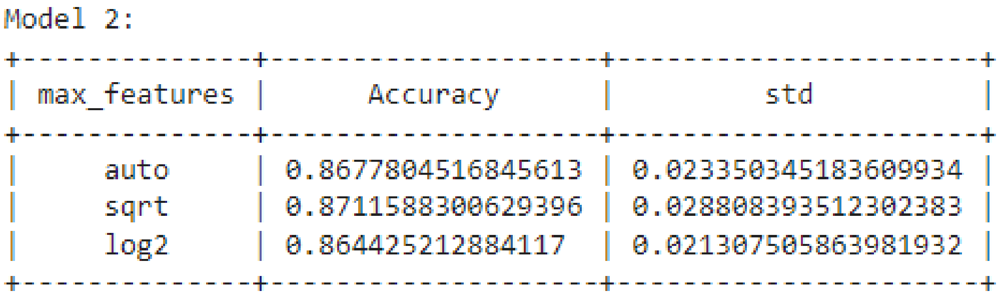
```
Model 1:
+--------------+--------------------+----------------------+
| n_estimators |      Accuracy      |         std          |
+--------------+--------------------+----------------------+
|      10      | 0.8270316549426139 | 0.02982302021124012  |
|      20      | 0.8490836727138097 | 0.021299626410377144 |
|      30      | 0.8643557941503146 | 0.029697771001670082 |
|      40      | 0.857599037393558  | 0.027260858442597364 |
|      50      | 0.8542206590151795 | 0.030867329651380546 |
|      60      | 0.8694696408737504 | 0.03108449842557713  |
|      70      | 0.8660681229174381 | 0.018743000991772358 |
|      80      | 0.8610005553498705 | 0.02426748273669653  |
|      90      | 0.8610468345057387 | 0.02304198923540557  |
|     100      | 0.8711819696408737 | 0.022485657895571602 |
|     110      | 0.8677341725286931 | 0.024496147814921383 |
+--------------+--------------------+----------------------+
```

We select **n_estimators=100**

Random Forest Classifier

Experiment : tune max_features



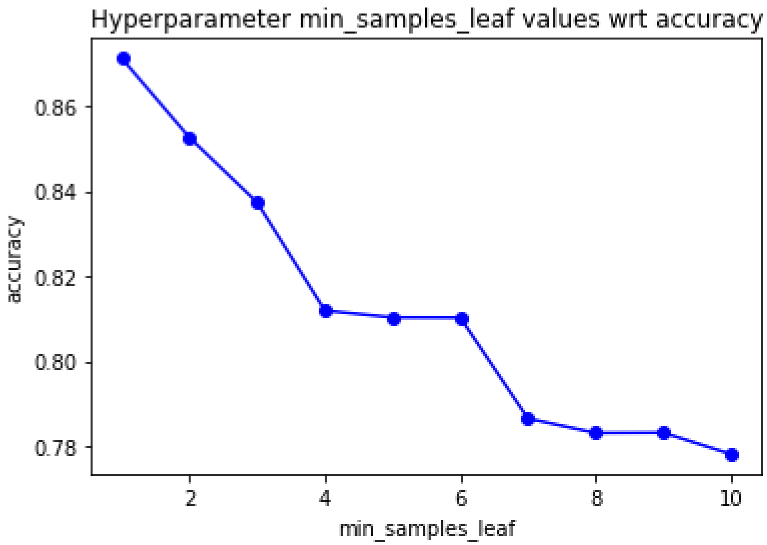


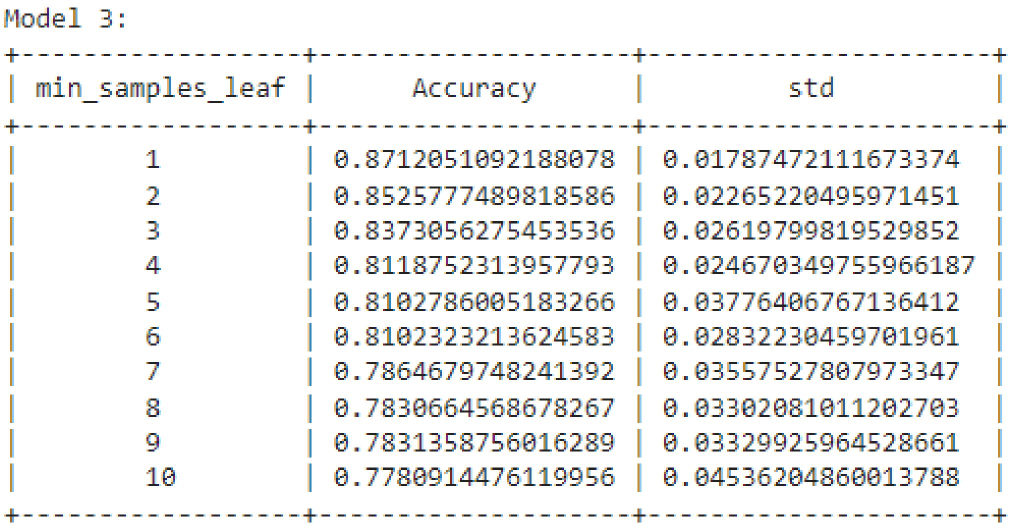
```
Model 2:
+---------------+---------------------+-----------------------+
| max_features  |      Accuracy       |          std          |
+---------------+---------------------+-----------------------+
|     auto      | 0.8677804516845613 | 0.023350345183609934 |
|     sqrt      | 0.8711588300629396 | 0.028808393512302383 |
|     log2      | 0.864425212884117   | 0.021307505863981932 |
+---------------+---------------------+-----------------------+
```

We select **max_features=sqrt**

Random Forest Classifier

Experiment : tune min_samples_leaf



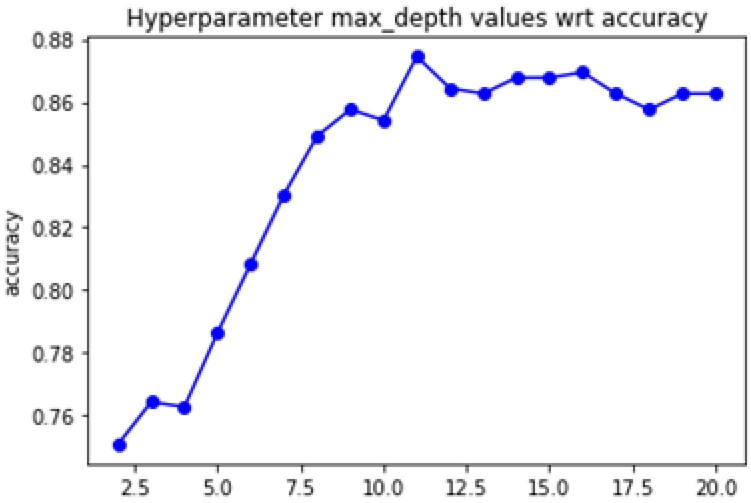


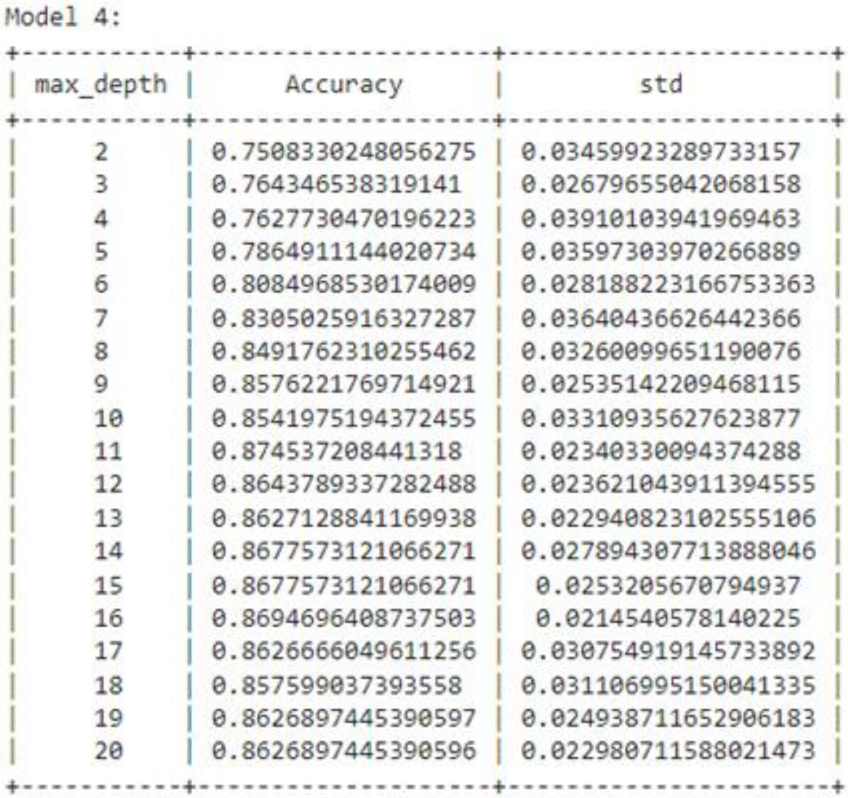
```
Model 3:
+------------------+--------------------+----------------------+
| min_samples_leaf |      Accuracy      |         std          |
+------------------+--------------------+----------------------+
|        1         | 0.8712051092188078 | 0.01787472111673374  |
|        2         | 0.8525777489818586 | 0.02265220495971451  |
|        3         | 0.8373056275453536 | 0.02619799819529852  |
|        4         | 0.8118752313957793 | 0.024670349755966187 |
|        5         | 0.8102786005183266 | 0.03776406767136412  |
|        6         | 0.8102323213624583 | 0.02832230459701961  |
|        7         | 0.7864679748241392 | 0.03557527807973347  |
|        8         | 0.7830664568678267 | 0.03302081011202703  |
|        9         | 0.7831358756016289 | 0.03329925964528661  |
|        10        | 0.7780914476119956 | 0.04536204860013788  |
+------------------+--------------------+----------------------+
```

We select **min_samples_leaf=1**

Random Forest Classifier

Experiment : tune max_depth



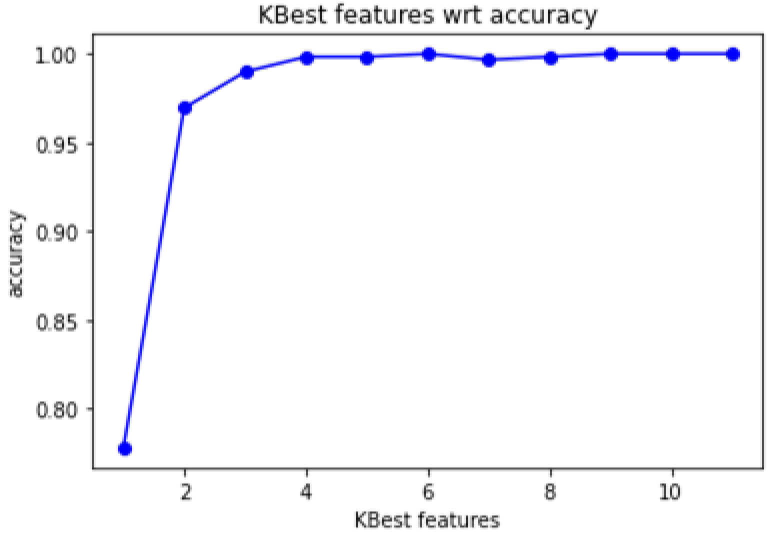


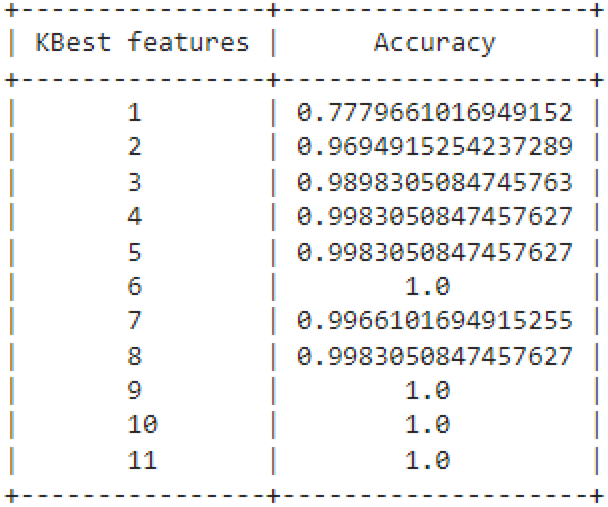
```
Model 4:

+-----------+--------------------+----------------------+
| max_depth |      Accuracy      |          std         |
+-----------+--------------------+----------------------+
|     2     | 0.7508330248056275 | 0.03459923289733157  |
|     3     | 0.764346538319141  | 0.02679655042068158  |
|     4     | 0.7627730470196223 | 0.03910103941969463  |
|     5     | 0.7864911144020734 | 0.03597303970266889  |
|     6     | 0.8084968530174009 | 0.028188223166753363 |
|     7     | 0.8305025916327287 | 0.03640436626442366  |
|     8     | 0.8491762310255462 | 0.03260099651190076  |
|     9     | 0.8576221769714921 | 0.02535142209468115  |
|     10    | 0.8541975194372455 | 0.03310935627623877  |
|     11    | 0.874537208441318  | 0.02340330094374288  |
|     12    | 0.8643789337282488 | 0.023621043911394555 |
|     13    | 0.8627128841169938 | 0.022940823102555106 |
|     14    | 0.8677573121066271 | 0.027894307713888046 |
|     15    | 0.8677573121066271 |  0.0253205670794937  |
|     16    | 0.8694696408737503 |  0.0214540578140225  |
|     17    | 0.8626666049611256 | 0.030754919145733892 |
|     18    | 0.857599037393558  | 0.031106995150041335 |
|     19    | 0.8626897445390597 | 0.024938711652906183 |
|     20    | 0.8626897445390596 | 0.022980711588021473 |
+-----------+--------------------+----------------------+
```

We select **max_depth=14**

Random Forest Classifier

Experiment : select k Best Features



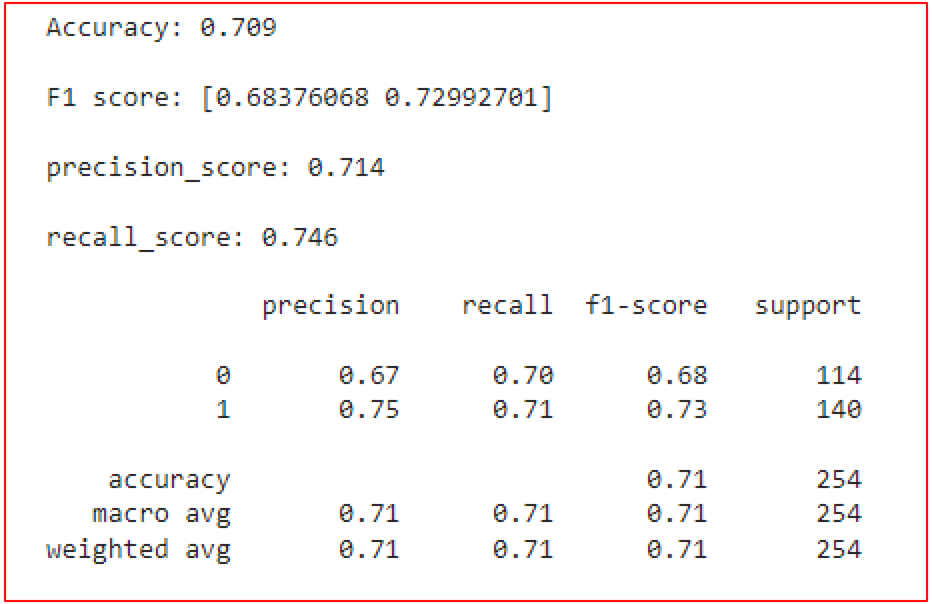


```
+----------------+--------------------+
| KBest features |      Accuracy      |
+----------------+--------------------+
|       1        | 0.7779661016949152 |
|       2        | 0.9694915254237289 |
|       3        | 0.9898305084745763 |
|       4        | 0.9983050847457627 |
|       5        | 0.9983050847457627 |
|       6        |        1.0         |
|       7        | 0.9966101694915255 |
|       8        | 0.9983050847457627 |
|       9        |        1.0         |
|       10       |        1.0         |
|       11       |        1.0         |
+----------------+--------------------+
```

We select **k=8**

Random Forest Classifier

Test: select k Best Features

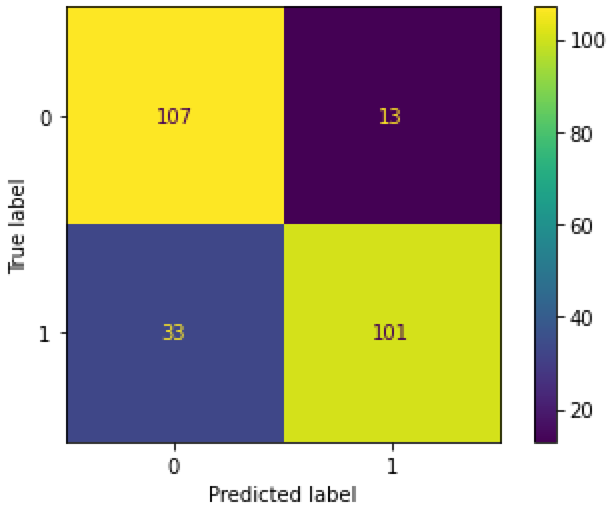


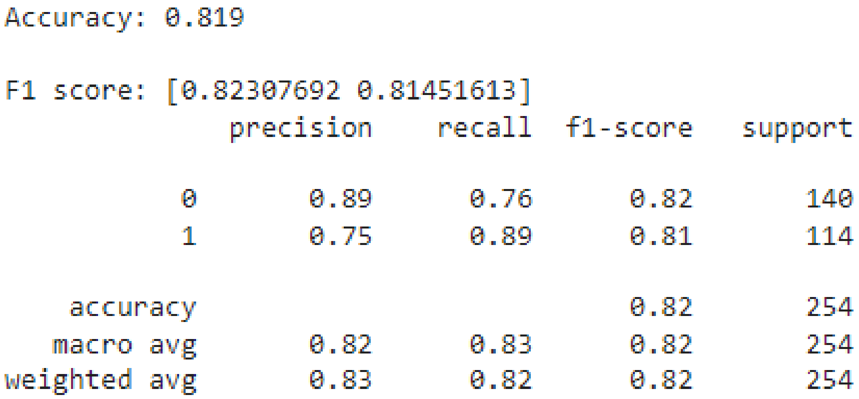
```
Accuracy: 0.709

F1 score: [0.68376068 0.72992701]

precision_score: 0.714

recall_score: 0.746

              precision    recall  f1-score   support

           0       0.67      0.70      0.68       114
           1       0.75      0.71      0.73       140

    accuracy                           0.71       254
   macro avg       0.71      0.71      0.71       254
weighted avg       0.71      0.71      0.71       254
```

Random Forest Classifier

Evaluation : criterion(Gini), n_estimators(100), min_samples_leaf(1), max_depth(14), max_features(sqrt)

```
Confusion Matrix:
```

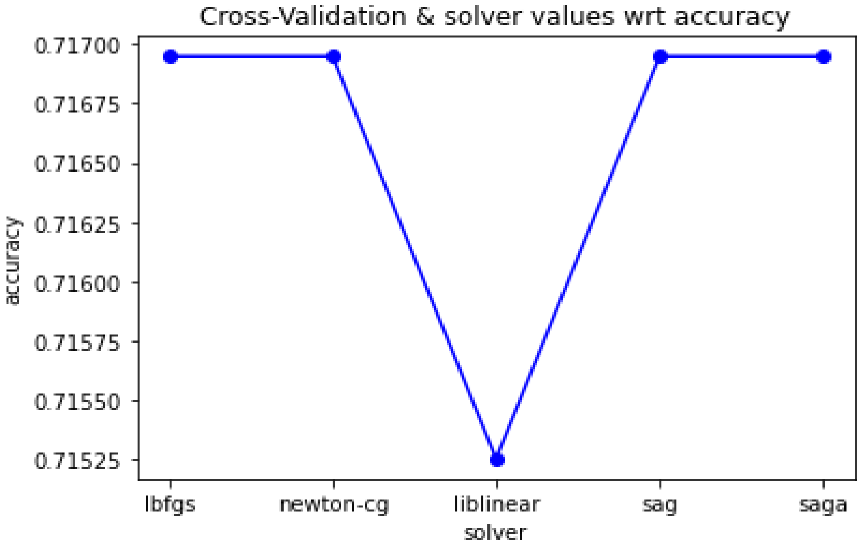


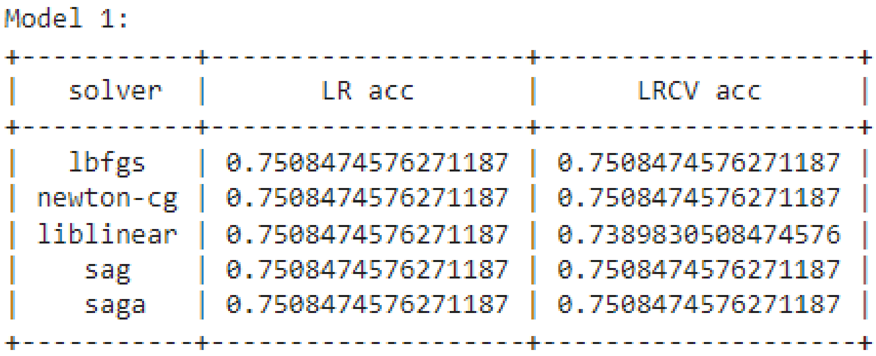


```
Accuracy: 0.819

F1 score: [0.82307692 0.81451613]
              precision    recall  f1-score   support

           0       0.89      0.76      0.82       140
           1       0.75      0.89      0.81       114

    accuracy                           0.82       254
   macro avg       0.82      0.83      0.82       254
weighted avg       0.83      0.82      0.82       254
```

Logistic Regression

Experiment - different solvers



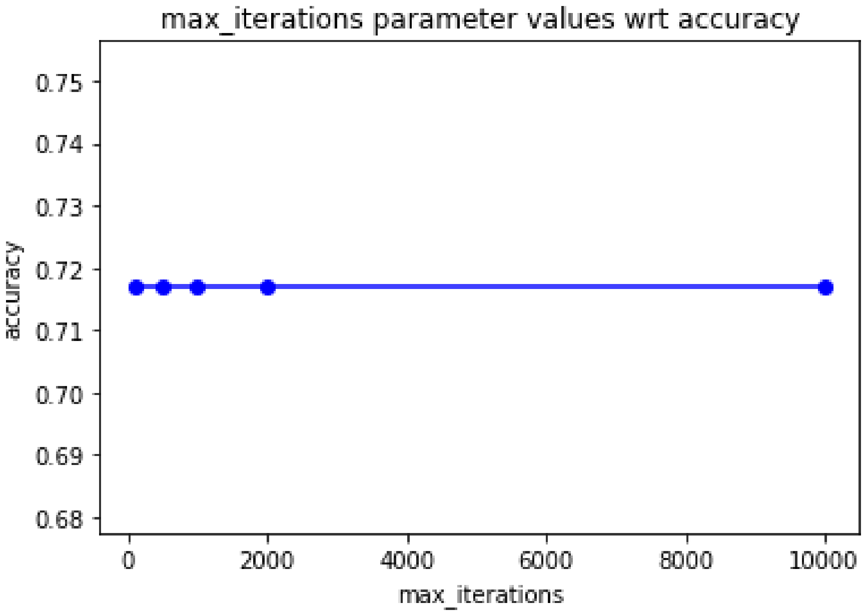


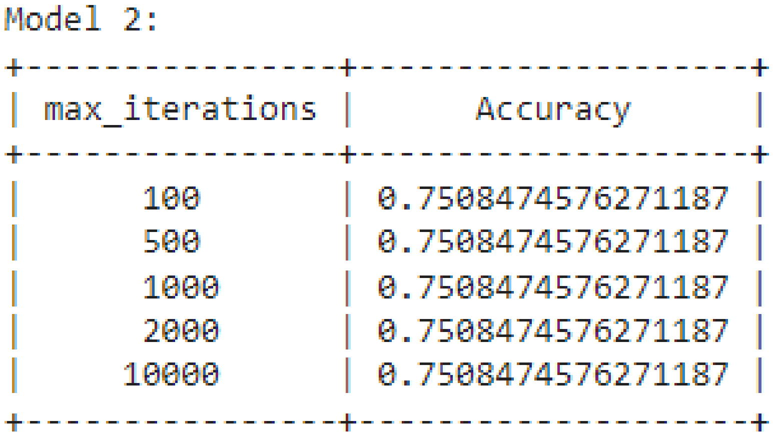
```
Model 1:
+-----------+--------------------+--------------------+
|   solver  |       LR acc       |      LRCV acc      |
+-----------+--------------------+--------------------+
|   lbfgs   | 0.7508474576271187 | 0.7508474576271187 |
| newton-cg | 0.7508474576271187 | 0.7508474576271187 |
| liblinear | 0.7508474576271187 | 0.7389830508474576 |
|    sag    | 0.7508474576271187 | 0.7508474576271187 |
|    saga   | 0.7508474576271187 | 0.7508474576271187 |
+-----------+--------------------+--------------------+
```

Since our data set is small, ‘newton-cg’ and 'lbfgs' are appropriate to use. We decided to choose **'lbfgs'**, the default solver, because it only stores the last few updates and saves memory.

Logistic Regression

Experiment - max_iterations parameter



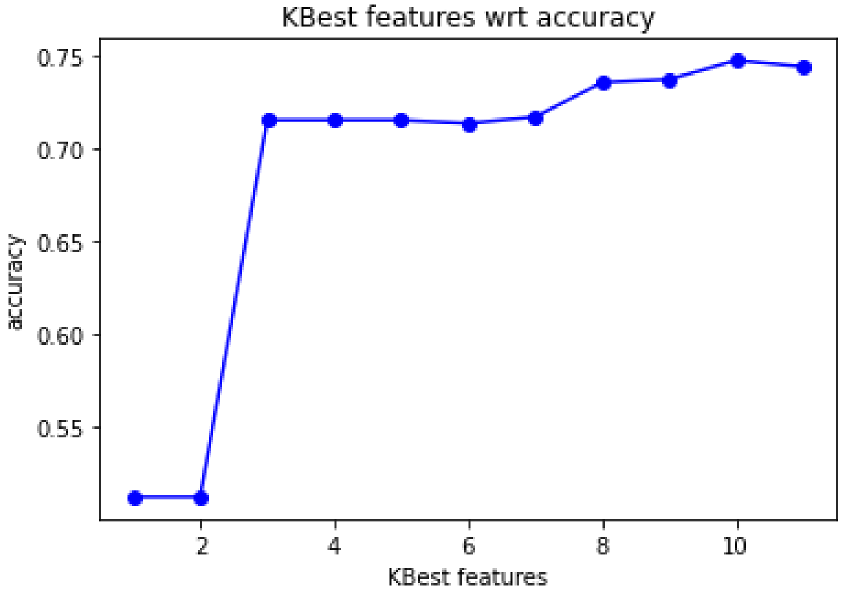


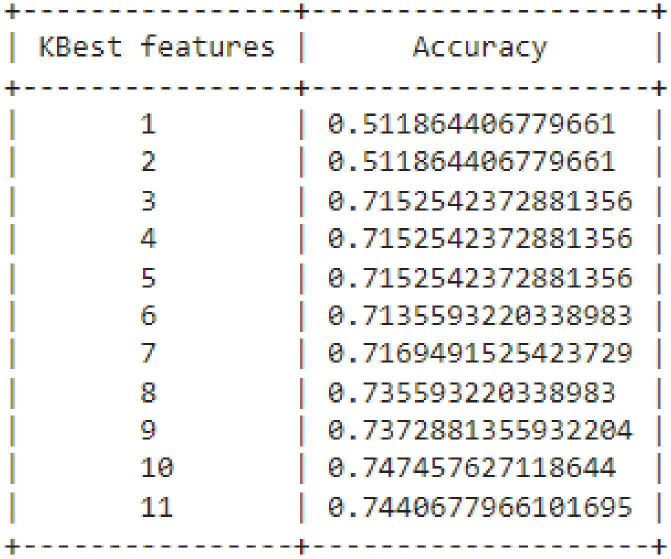
```
Model 2:
+----------------+---------------------+
| max_iterations |       Accuracy      |
+----------------+---------------------+
|      100       | 0.7508474576271187  |
|      500       | 0.7508474576271187  |
|      1000      | 0.7508474576271187  |
|      2000      | 0.7508474576271187  |
|     10000      | 0.7508474576271187  |
+----------------+---------------------+
```

No difference was found.

Logistic Regression

Experiment : k-best features





```
+----------------+---------------------+
| KBest features |       Accuracy      |
+----------------+---------------------+
|       1        |  0.511864406779661  |
|       2        |  0.511864406779661  |
|       3        | 0.7152542372881356  |
|       4        | 0.7152542372881356  |
|       5        | 0.7152542372881356  |
|       6        | 0.7135593220338983  |
|       7        | 0.7169491525423729  |
|       8        |  0.735593220338983  |
|       9        | 0.7372881355932204  |
|       10       |  0.747457627118644  |
|       11       | 0.7440677966101695  |
+----------------+---------------------+
```

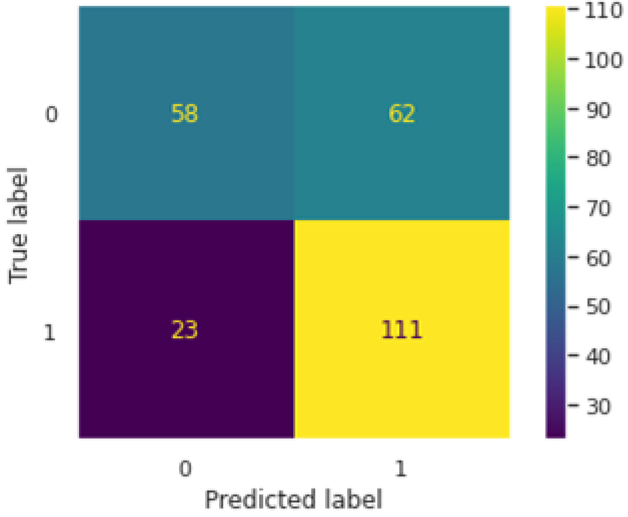
No improvement was found

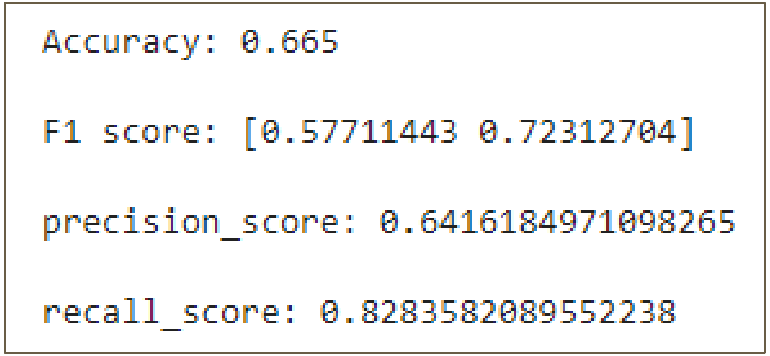
Logistic Regression

Evaluation

Since no improvement was found, we keep the default model.

```
Confusion Matrix:
```





```
Accuracy: 0.665

F1 score: [0.57711443 0.72312704]

precision_score: 0.6416184971098265

recall_score: 0.8283582089552238
```

Best Model

Model	Accuracy	Recall
Decision Tree	0.7	0.7
Random Forest	0.82	0.82
Logistic Regression	0.67	0.82