

Small Business Loans: Predicting Default

...

Improving The Small Business Administration Loan
Commitment Program

Dillan Gump

Context: Small Business Administration Loan Program

- SBA sponsors a program that commits to guarantee a portion of a loan
- If a loan defaults, the SBA is responsible for the portion committed
- If only there was a way to know if a loan would default ahead of time...

Purpose - For the SBA

- Help the SBA decide which loans to approve
- Determine the factors that contribute to a loan defaulting
- Improve upon traditional selection techniques with machine learning

What will a good model look like?

- Approves as many loans as possible while minimizing risk
 - Avoid backing loans that are likely to default
- Prioritizes incorrectly declining incorrectly accepting
 - Conservative model
- Sacrifices and Trade-Offs
 - Will end up missing some loans that would not have defaulted
 - Big Con. Can be tuned.

What We Know

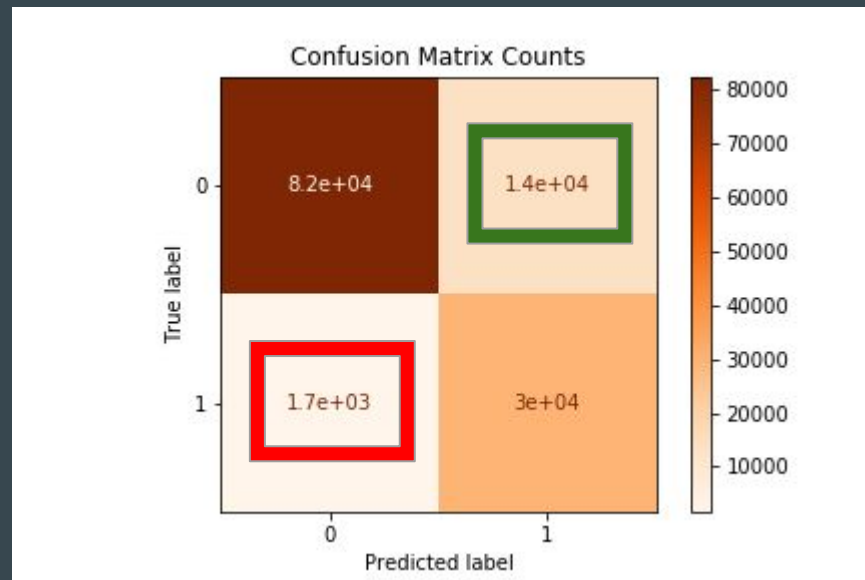
- Original Data
 - ~800,000 loans
 - 1965 - 2014
- Selected Data
 - ~500,000 loans
 - 2000 - 2014
 - Completed Loans Only
- Features
 - Loan Information
 - Bank Information
 - Business Information
 - SBA information

New Information

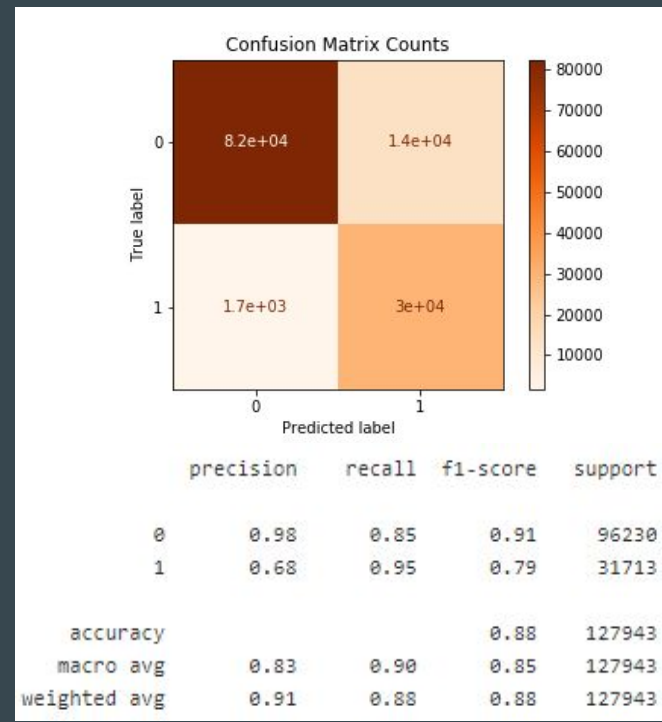
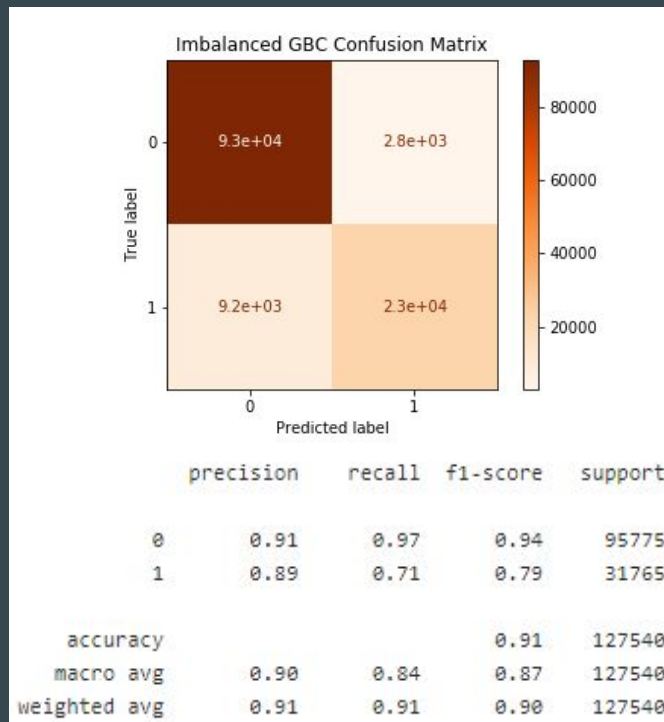
- Using the existing data, a few new features were created
- New Features:
 - Yes/No: Is the business a franchise?
 - Yes/No: Is the lending bank out of state?
 - Yes/No: Was the loan approved by the SBA before disbursement?
 - Levels: Number of jobs created and number of jobs retained
 - Levels: Bank size
 - Percentage: Portion of loan covered by SBA
- Good start towards model improvement

Model Summary

- Accuracy: 0.88
- Precision: 0.68
- Recall: 0.95
- Correctly identified more defaulted loans than paid loans



Imbalanced Classifier Comparison - Tuning Demo




```

1 preprocessing = ColumnTransformer(
2     [
3         ("leaveOneOut", LeaveOneOutEncoder(), cat_cols),
4         ("scale", StandardScaler(), num_cols), # never hurts
5         # ("knnImpute", KNNImputer(n_neighbors=2), impute_cols),
6         # ("simpleImpute", SimpleImputer(), impute_cols),
7     ],
8     remainder="passthrough",
9 )

```

```

1 n_trees = 100
2 learning_rate = 2 / n_trees
3
4 pipeline = Pipeline(
5     [
6         ("preprocessing", preprocessing),
7         ("xgbClass", XGBClassifier(n_estimators=n_trees, learning_rate=learning_rate)),
8     ]
9 )

```

```

1 grid = {
2     "xgbClass__subsample": [0.00125, 0.0025, 0.01],
3     # "gbr__max_features": [0.5, 0.75, 1.0], # alternative
4     "xgbClass__colsample_bytree": [0.6, 0.8, 1.0],
5     "xgbClass__max_depth": [4, 6, 7, 8],
6 }
7
8

```

1 pipeline_cv_reclean.best_params_

```

{'xgbClass__colsample_bytree': 1.0,
 'xgbClass__max_depth': 6,
 'xgbClass__subsample': 0.01}

```

```

1 resample_grid = {
2     "xgbClass__subsample": [0.1, 0.5],
3     "xgbClass__colsample_bytree": [0.6, 0.8, 1.0],
4     "xgbClass__max_depth": [4, 6, 7, 8],
5 }
6

```

1 pipeline_cv_resample_reclean.best_params_

```

{'xgbClass__colsample_bytree': 1.0,
 'xgbClass__max_depth': 7,
 'xgbClass__subsample': 0.1}

```

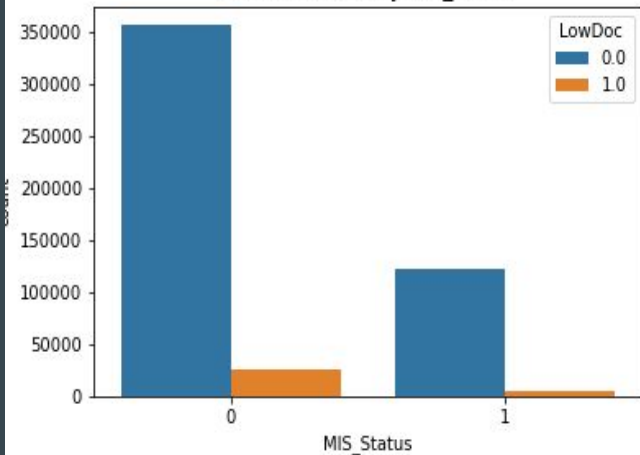
Resampling Technique

```
1 X_train_0 = X_train[y_train == 0]
2 X_train_1 = X_train[y_train == 1]
3
4 n_0 = X_train_0.shape[0]
5 n_1 = X_train_1.shape[0]
6
7 # Sample majority class to have less observations
8 X_train_0_sample = X_train_0.sample(n_1, replace=False, random_state=42)
9
10 ## Sample minority class to have less observations
11 # X_train_1_sample = X_train_1.sample(n, replace=True, random_state=42)
12
13 X_train_resample = pd.concat((X_train_1, X_train_0_sample))
14 X_train_resample = X_train_resample.reset_index(drop=True)
15
16 y_train_resample = np.array([1] * n_1 + [0] * n_1)
17 y_train_resample.mean()
```

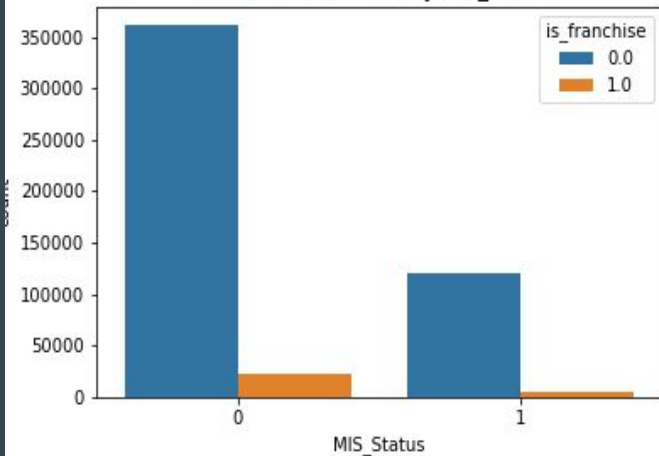
Important Features

- Low Documentation (Y/N): Whether a loan under \$150k can complete an alternative 1-page application
 - Strongest predictor. Makes sense as it is a kind of prescreening
- Job Category(Levels): How many jobs were created
 - Loans in the 10-100 jobs created range had the smallest percentage default
- Bank State (Categorical): State of bank issuing loan
- Disbursement Gross (Continuous): Defaulted loans tended to have a smaller sum
 - Mean and median both low
- Is Franchise (Y/N): Is the business a franchise?
- Bank Size (Levels): Based on number of loans given

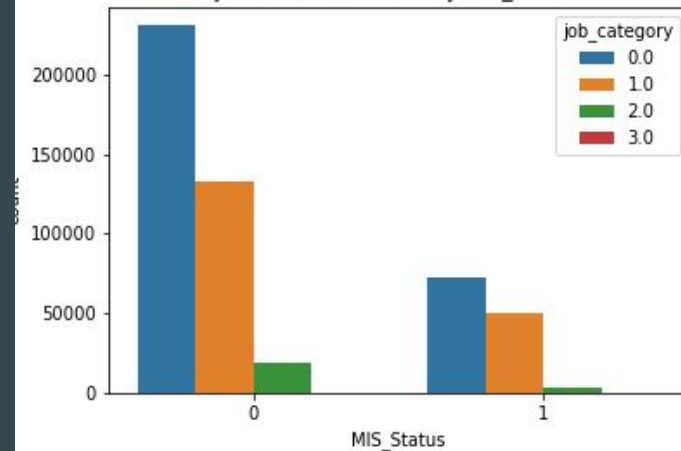
LowDoc Counts by MIS_Status



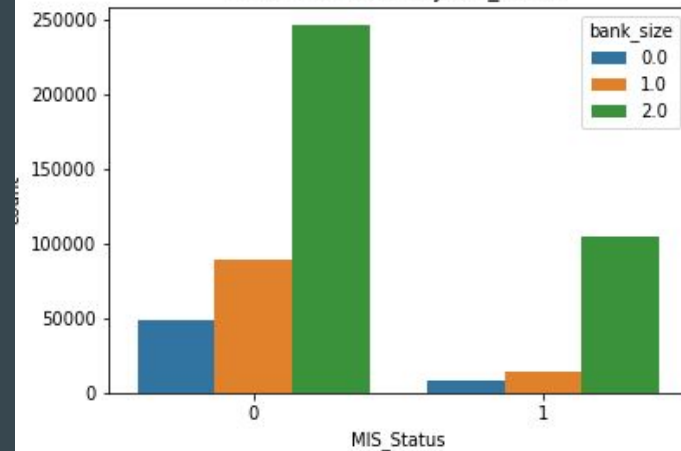
Is-Franchise Counts by MIS_Status



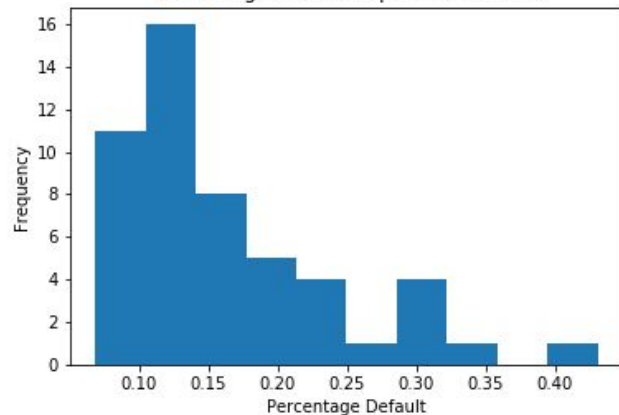
Jobs Created Counts by MIS_Status



Bank Size Counts by MIS_Status



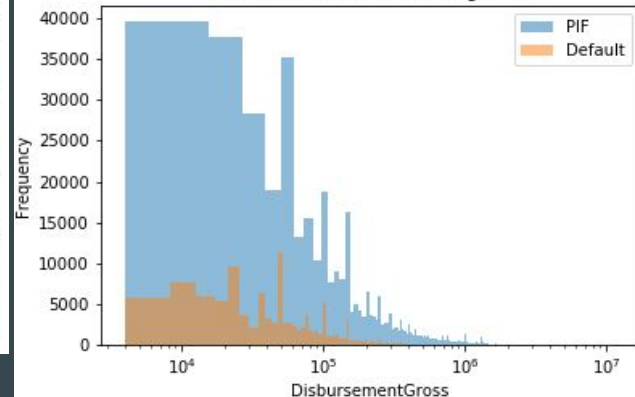
Percentage Defaulted per Banks State



BankState

VA	0.430738	OK	0.137051
NC	0.339468	AK	0.134615
SD	0.316343	TN	0.130936
CA	0.307649	MS	0.128300
OR	0.293199	NJ	0.126822
IL	0.291240	HI	0.126208
DE	0.285044	NV	0.125000
FL	0.244540	MI	0.123644
NY	0.237958	KS	0.123408
RI	0.232725	MD	0.120241
UT	0.219884	WA	0.117607
OH	0.204191	MN	0.114859
SC	0.202687	IN	0.112768
AL	0.199076	LA	0.107488
CT	0.190476	NM	0.102632
TX	0.180539	PA	0.096965
MO	0.175309	ME	0.095610
WI	0.168798	MA	0.094795
WV	0.167010	AZ	0.089459
DC	0.162338	ND	0.083294
IA	0.162050	NH	0.082742
AR	0.161411	CO	0.076905
KY	0.159182	MT	0.069466
GA	0.159182	VT	0.069134
ID	0.139535	WY	0.068323
NE	0.138377		

Gross Disbursement on Log Axes



Final Interpretation

- Existing signs of confidence are good indicators of loan success
 - Approved for Low Doc, growing employees
- Not really any harm in losing precision
 - Those loans ended up paid anyways
- The data shows a lot of potential
 - True effect of state
 - Refine engineered features
-

Feature	Importance
LowDoc	0.280424
job_category	0.133911
BankState	0.078828
DisbursementGross	0.055349
is_franchise	0.051034
bank_size	0.047374
State	0.042923
RevLineCr	0.034007
bank_out_of_state	0.032604
UrbanRural_cleaned	0.031403
Disbr_year	0.030572
twoDigNAICS	0.028628
NewExist	0.027750
retained_category	0.025924
NoEmp	0.025847
Term_years	0.025280
Disbr_Month_cos	0.024222
Disbr_Month_sin	0.023920

LowDoc	0.0	1.0
MIS_Status		
0	0.746469	0.847128
1	0.253531	0.152872

	DisbursementGross	
	mean	median
LowDoc		
0.0	192753.906139	75000.0
1.0	86988.485540	82300.0

	DisbursementGross	
	mean	median
MIS_Status		
0	209783.519929	87300.0
1	115712.153896	51437.0

		DisbursementGross	
		mean	median
LowDoc	MIS_Status		
0.0	0	218678.793757	88599.0
	1	116630.470726	50000.0
1.0	0	86153.722524	80000.0
	1	91616.073843	90000.0

	DisbursementGross	
	mean	median
bank_size		
0.0	260532.150587	132000.0
1.0	294007.328827	150000.0
2.0	142883.929899	51243.5

job_category	0.0	1.0	2.0	3.0
MIS_Status				
0	0.760818	0.725964	0.843943	0.76584
1	0.239182	0.274036	0.156057	0.23416

is_franchise	0.0	1.0
MIS_Status		
0	0.749927	0.796707
1	0.250073	0.203293

	DisbursementGross	
	mean	median
job_category		
0.0	166810.504676	66808.0
1.0	178753.683634	79100.0
2.0	506060.262202	350000.0
3.0	601604.148760	393000.0

Implementation

- Long Term Validation Study
- Deploy and compare several new strategies
 - Control- Traditional methods
 - Other models built with different information
 - Collaboration with business team
 - Introduce new data, engineer new features
- Integrate model as a software tool
 - Possible automation
 - Augment Existing Process
- Supplement with Diversity and Inclusion Measures
 - Model won't challenge historical trends

Questions?

Thank You!

Appendix

- SBA Link: <https://www.sba.gov/offices/headquarters/ofa/resources/11421>
- Link to original Kaggle set:
<https://www.kaggle.com/mirbektoktogaraev/should-this-loan-be-approved-or-denied>
-