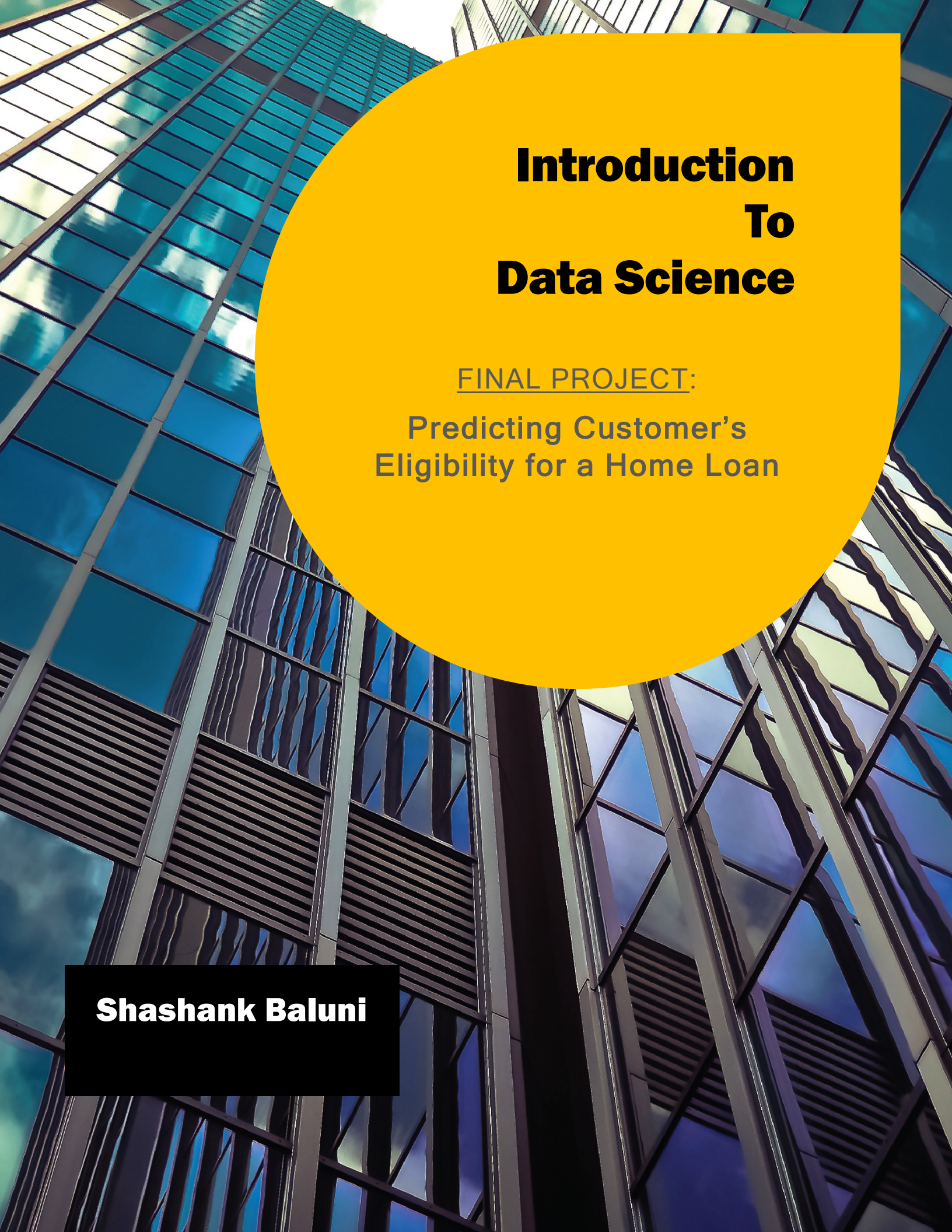


Introduction To Data Science

FINAL PROJECT:

Predicting Customer's Eligibility for a Home Loan

Shashank Baluni

Introduction

In an era where real estate is a cornerstone of financial security, access to housing loans plays a pivotal role in fulfilling the dreams of countless individuals and families. However, the process of determining loan eligibility involves a multitude of factors, from financial history to employment status and creditworthiness. This project endeavors to harness the power of data science to streamline and enhance this crucial decision-making process.

Background

Obtaining a housing loan is a significant financial milestone, enabling individuals to make substantial investments in real estate. Lenders, on their part, face the challenge of assessing the risk associated with each applicant. Traditionally, this has been a time-consuming process, often reliant on manual evaluation and subjective judgment. With advancements in data science and machine learning techniques, it is now possible to leverage historical data to develop models that can predict loan eligibility.

Motivation

The motivation behind this project stems from the desire to democratize access to housing loans, making it more efficient and equitable for all prospective homeowners. By employing predictive modeling, I aim to create a tool that can swiftly and objectively evaluate loan applications, providing financial institutions with a reliable mechanism to assess risk while ensuring deserving applicants receive the support they need.

Research Question

The central question driving this study is: “Can a data-driven approach accurately predict the eligibility of an individual for a housing loan based on a set of relevant features”? By employing machine learning algorithms on historical loan data, I seek to develop a model that can efficiently and reliably determine whether an applicant meets the criteria for a housing loan.

Data Summary

Data Source: https://www.kaggle.com/datasets/vikasukani/loan-eligible-dataset

About Data Variables:

A company wants to automate the loan eligibility process (real-time) based on customer details provided while filling the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others. To automate this process, I have been given a problem to identify the customer's segments, those who are eligible for loan amount so that the company can specifically target these customers. Here I have partial data set with below columns:

Variable Name	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)

Dependents	Number of dependents
Education	Applicant Education (Graduate/ Under-Graduate)
Self_Employed	Self-employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Co-applicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of a loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi-Urban/ Rural
Loan_Status	Loan approved (Y/N)

**Variables in bold are selected to run Logistic Regression model(s)*

Statistical Summary: The data contains 614 observations across 13 columns.

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614	614	592	600	564
mean	5403.45928	1621.2458	146.412162	342	0.842199
std	6109.04167	2926.24837	85.587325	65.12041	0.364878
min	150	0	9	12	0
25%	2877.5	0	100	360	1
50%	3812.5	1188.5	128	360	1
75%	5795	2297.25	168	360	1
max	81000	41667	700	480	1

Frequency Distribution of some categorical variables:

Gender

- Male: 489
- Female: 112

Married

- Yes: 398
- No: 213

Education

- Graduate: 480
- Not Graduate: 134

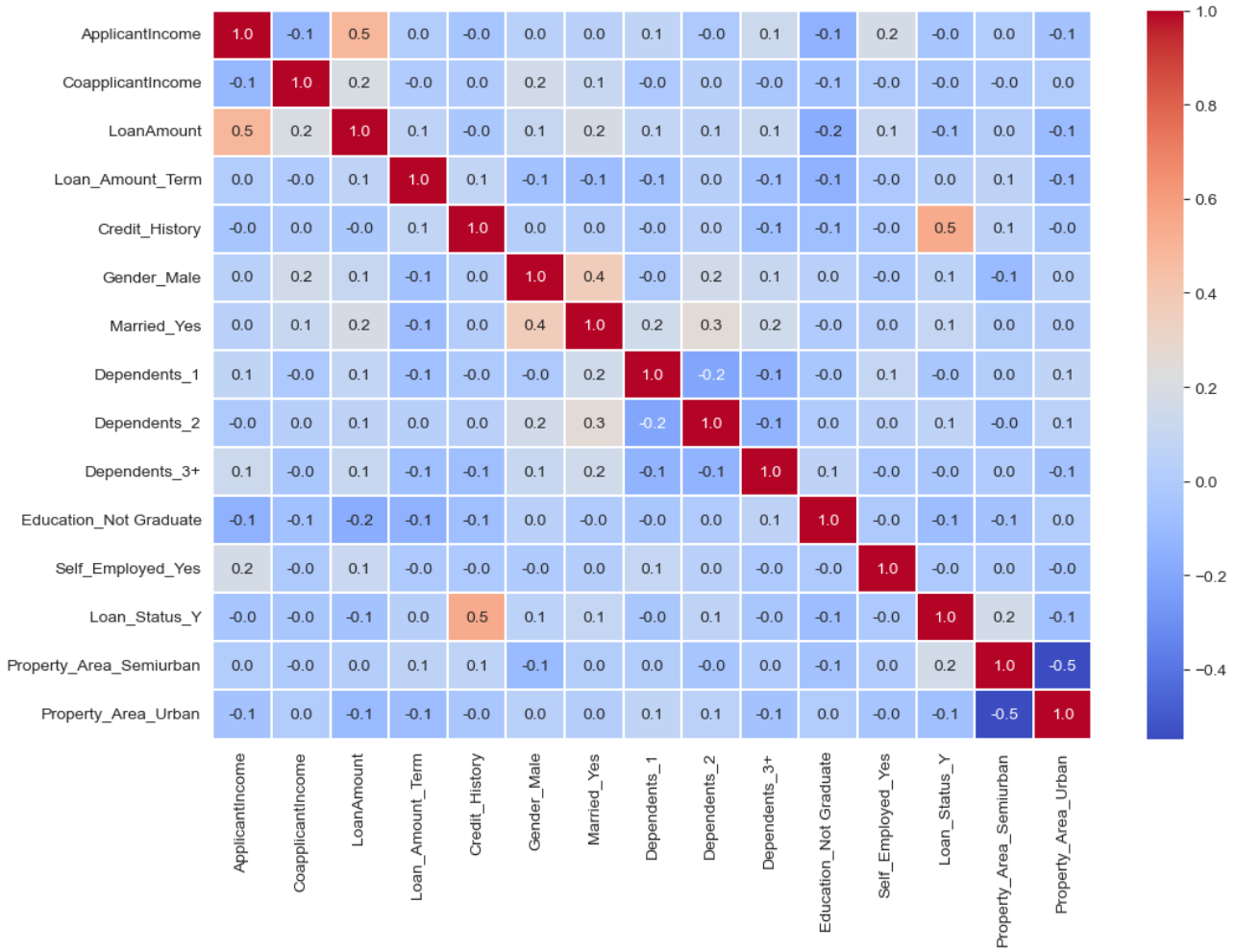
Loan Status

- Y: 422
- N: 192

Data Cleaning:

For data cleaning, I first checked the fill rate of each column. Any column with missing values should either be removed or replaced with a proper value (mean or mode of the variable). I then used One-Hot Encoding to convert any important Categorical variable into Numerical values, like: Loan_Status. The process of encoding categorical variables is crucial because many machine learning algorithms, especially those in scikit-learn, work with numerical data. Encoding transforms categorical values into a format that these algorithms can understand and use for model training.

Correlation Matrix Across Variables:



ApplicantIncome and LoanAmount has high positive correlation of 0.5. Credit_History has a high correlation of 0.5 with Loan_Status. Other variables like Married and Gender; and Married and Dependents (2) seem to be somewhat correlated.

Methodology & Results

As I am trying to predict whether a customer will be eligible for a home loan or not, which is a class-based problem, I used Logistic Regression model. For example, a customer will be eligible for a home loan or not, where 0 means “Not Eligible” and 1 means “Eligible”.

About Logistic Regression:

Logistic Regression is a statistical method used for binary classification problems, where the outcome variable is categorical and has two classes. Despite its name, logistic regression is a classification algorithm and not a regression algorithm. It is particularly useful when the dependent variable is binary, representing two classes like 'Yes' or 'No,' '1' or '0,' or 'True' or 'False'.

Use Cases:

- Binary Classification: Logistic Regression is widely used for binary classification tasks such as spam detection, disease diagnosis, and loan approval prediction.
- Probabilistic Output: It's suitable when there's a need for probabilistic outputs, allowing for a nuanced interpretation of predictions.

Model Variables:

For this classification problem, I used Credit_History, Education, ApplicantIncome and Gender as our input variables and Loan_Status as the output variable which has two classes - 1 for eligible and 0 for not-eligible. These input variables do not have any high correlation with each other but seem to be somewhat correlated to output variable.

Model Equation:

The logistic model is formulated using the probability (p) of a categorical variable as dependent variable, and a series of variables as independent variables:

$$\ln\left[\frac{p}{1-p}\right] = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon$$

where p is the probability that event A happens which is "Loan_Status" is 1 (eligible). Here, X_1 is Credit_History variable, X_2 is Education variable, X_3 is ApplicantIncome variable and X_4 is Gender variable. We call $\frac{p}{1-p}$ as the "odds ratio" – how likely the event will happen.

Model Results:

Here are the results of two different iterations of Logistic regression models using different set of input variables and same output variable "Loan_Status":

Logistic Regression Results Comparison

	Model 1	Model 2
Intercept	-2.74 ***	-2.88 ***
Credit_History	3.99 ***	3.87 ***
Education (Not Graduate)	-0.41	-0.37
ApplicantIncome	-2.9e-05	-
Gender (Male)	0.45	0.44
No. of Observations	408	408

**p < 0.05, **p < 0.01, ***p < 0.001*

The first value indicates coefficients, and star (*) indicates the significant level (p-value) of the variable.

Based on the results from model 1, ApplicantIncome variable's p-value is significantly greater than the threshold, hence, I ran another model (Model 2) without this variable which further improved train model accuracy and results. Only Credit_History variable has its p-value less than threshold; hence, this variable is significant in explaining Loan_Status. Credit_History and Gender_Male variables are positively associated with the target variable, however, Education_Not Graduate is negatively related. As my final model, I chose model 2.

Model Accuracy:

With Model 2, I attained 80.4% train model accuracy and 82.5% accuracy rate on testing data. This means approximately 82.5% of the predictions made by the model are correct.

With these coefficients from Model 2, I found predicted values of the output variable for my test dataset and ran a confusion matrix to evaluate the results.

Confusion Matrix:

		Actual Values	
		Negative (0)	Positive (1)
Predicted Values	Negative (0)	17 (TN)	3 (FN)
	Positive (1)	15 (FP)	68 (TP)

The confusion matrix provides a detailed breakdown of the model's performance. Here's an interpretation based on the provided confusion matrix:

- True Negative (TN): Instances that are actually class '0' and correctly predicted as class '0'.
- False Positive (FP): Instances that are actually class '0' but incorrectly predicted as class '1'.
- False Negative (FN): Instances that are actually class '1' but incorrectly predicted as class '0'.
- True Positive (TP): Instances that are actually class '1' and correctly predicted as class '1'.

The model correctly predicted 17 instances of loan rejection and 68 instances of loan approval. The model missed 3 instances of actual loan approval and incorrectly predicted 15 instances as loan approval when they were actually loan rejections.

Model Performance:

The metrics provided below are part of the classification report, which is a summary of the performance of a classification model. Let's break down the key metrics:

class	precision	recall	f1-score	support
0	0.85	0.53	0.65	32
1	0.82	0.96	0.88	71

- Precision: A ratio of correctly predicted positive observations (True Positives) to the total predicted positives (True Positives + False Positives). When the model predicts class '0', it is correct 85% of the time.
- Recall: Recall, also known as Sensitivity or True Positive Rate, is the ratio of correctly predicted positive observations (True Positives) to the total actual positives (True Positives + False Negatives). The model captures only 53% of all instances that actually belong to class '0'.
- F1-Score: A weighted average of precision and recall. It is particularly useful when there is an uneven class distribution. It is 0.65 for class '0', providing a balance between precision and recall.
- Support: Number of actual occurrences of the class in the specified dataset. In this case, it indicates that there are 32 instances of class '0' in the dataset.

Conclusion & Future works

Conclusion:

The model is generally accurate, with a high precision for both class '0' and class '1' and a very high recall for class '1'. This means that the model is good at identifying instances of class '1', but its performance on class '0' is less satisfactory. This might be a point for improvement, depending on the specific goals of the model.

Model results show that Credit_History, Education and Gender can be used to predict whether the customer is eligible for home loan or not. This aligns with my assumptions that these variables should have high impact on loan eligibility. I had similar observations from Correlation matrix as well.

Limitations of Logistic regression model:

- Logistic Regression assumes linear relationship between input and dependent variables. This means that it may not be able to capture complex non-linear relationships.
- Logistic regression requires a relatively large sample size to ensure stable estimates of the model parameters. In situations with small sample sizes, the model may suffer from overfitting or underfitting.
- It cannot handle missing data well. Logistic regression requires complete data, and missing data can lead to biased or inaccurate results.
- Amongst other limitations, the model is sensitive to outliers, assumes independence of observations and absence of multicollinearity and is limited to binary outcomes.

Potential Scope:

- Prediction based models have high demand. Based on the results, companies can evaluate whether to give loan to a customer or not which can help with fraud prevention.
- As Logistic regression model has its limitations, one can explore other classification models like Decision Tree, SVM, etc.
- This project can further be extended to other areas like auto loan, student loan, etc. however, features should be reevaluated based on their significance on the output.