Don’t Overfit!

Hardik Gourisaria, Mayank Agarwal, Amit Kumar
PES1201700129, PES1201701349, PES1201701295

***Abstract*—This work introduces a model which when trained on a dataset of 300 features and 250 training samples, does not overfit the data set and provides good results for the testing dataset that has around 20,000 unlabeled testing samples. This work comprises two main aspects. They are feature engineering and classification.**

***Index Terms*—Feature Engineering, Overfitting, Component Analysis, Classification**

I. Introduction

THIS work aims to develop a model that relies on the concepts of feature engineering and classification to fit a model to a dataset with 300 features and 250 training samples such that the model does not overfit [3] and provides high accuracy results for 19750 unlabeled testing samples.

The problem is picked up from a Kaggle Competition called “Don’t Overfit!” [1] and hence the title.

Two main aspects to be considered for solving the problem to be considered are Feature [5] Engineering and Classification. Feature Engineering is required so as to reduce the number of features in the dataset, thus reducing overfitting. It also helps us consider strictly only the features that are related to the desired output. This is an essential step and the reason has been discussed in the following sections. Classification is required because the problem is essentially a classification problem where we have to predict the class levels of a sample based on the feature values provided.

II. Feature Engineering

A. What is it?

Feature engineering [5] is the process of using domain knowledge to construct a dataset with features pertaining to the required problem statement. Performing feature engineering makes the Machine Learning models work well. If features are not picked properly, a model trained on the dataset may not yield desirable results due to redundancies and overfitting. The basic essence is to make life easier for the machine learning model developer as picking the right and relevant features simplifies the understanding phase, development phase and deployment phase of the model. This provides better results for the addressed problem with shorter development time and expense.

B. Techniques Involved

Feature engineering can be used to address many problems in the dataset such as fixing faulty data, replacing or removing missing values, dimensionality reduction [2], etc. In this work, dimensionality reduction is by far the most important aspect of feature engineering that is essential to develop the machine learning model that performs well. Given the training and testing datasets have 300 features, it is necessary to reduce the number of features so as to reduce the model complexity, improve understandability, reduce development time and reduce the chances of overfitting etc. to name a few. Principle Component Analysis is one of the dimensionality reduction techniques that seems to be appropriate to be used. It has been explained briefly in the following sections.

III. Overfitting

A. What is it?

Overfitting [3] is a situation where the model provides high accuracy on the training dataset, however fails to do on the testing dataset. It can also be said that the model learns the training data too well and is unable to generalize well on the testing dataset. An overfitted model is a statistical model that contains more parameters than can be justified by the data. In the case of our work, the model has a high chance of overfitting as there are 300 features that the model can be trained on. This leads to the increase in complexity of the model and hence overfitting.

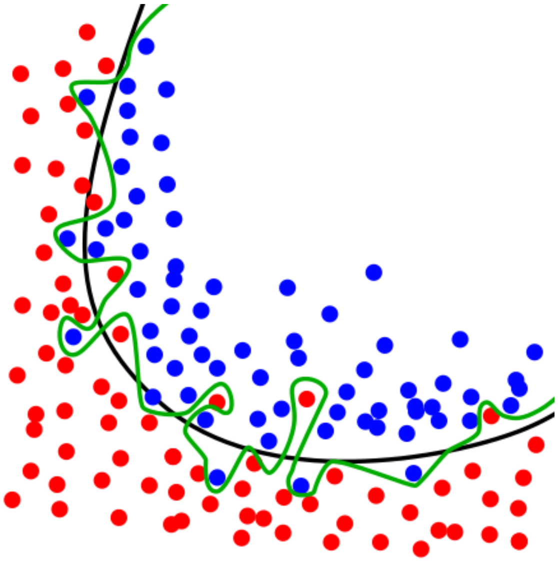


Fig 1. Diagram comparing a overfitted model against a well fitted model. The green line represents the overfitted model and the black line represents the well fitted model.

It can also be said that when the model is trained on many features, it starts to learn the noise in the data too and then does not provide correct predictions because it focusses too much on the moisy details in the data.

B. *Reducing Overfitting*

Some of the common ways to reduce overfitting [4] are to use methodologies like cross validation, early stopping, pruning, regularization, dimensionality reduction etc. just to name a few. In our work the main focus is on using dimensionality reduction to reduce the number of parameters in the model and hence reducing the complexity and chance of overfitting.

IV. Dimensionality Reduction

A. *What is it?*

In machine learning problems, there are often too many variables called features based on which the prediction is performed. The higher the number of features, the harder it gets to visualize the training dataset and come up with appropriate algorithms to solve the problem. Sometimes, the features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction reduces the number of features under consideration, by picking only the relevant features or dropping the redundant features.

B. *Feature Selection*

In this, we try to obtain a subset of the original feature set from the dataset, to model the problem. It usually involves three ways:

1. Filter
2. Wrapper
3. Embedded

C. *Feature Extraction*

This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

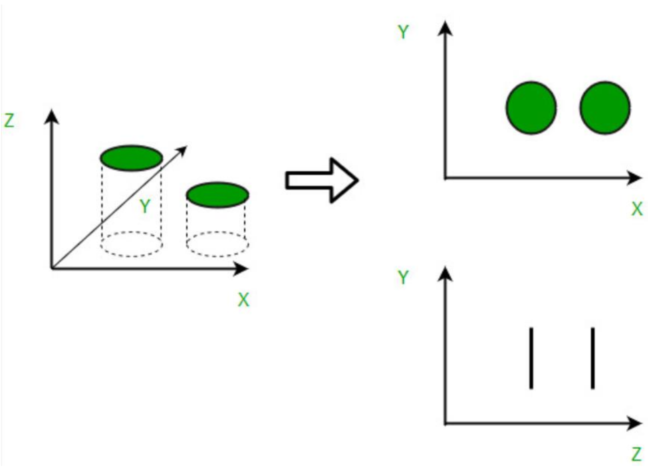


Fig 2. Diagram depicting dimensionality reduction by feature extraction. The 3D model is reduced to a 2D model which is easier to understand by the removal of one axis or feature

D. *Methods*

The various methods used for dimensionality reduction include:

1. Principal Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)
3. Generalized Discriminant Analysis (GDA)

Dimensionality reduction may be both linear or non-linear, depending upon the method used.

E. *Advantages*

- It helps in data compression, and hence reduced storage space.
- It reduces training and testing time and saves cost.
- It also helps remove redundant features, if any and hence reduces chances of overfitting hence improving the model performance.

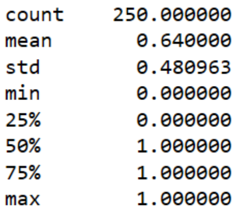
F. *Disadvantages of Dimensionality Reduction*

- Important details may be lost.
- In the case of PCA, it finds linear correlations between features which is undesirable sometimes
- How many principle components are to be considered may not be easy to determine and some basic thumb rules may need to be followed.

V. Exploratory Data Analysis

Exploratory Data Analysis was performed on the dataset and the following inferences were made:

1. The data was found to be clean with no missing values.
2. The dataset is used for a binary classification problem and following are the statistics of the target values.



```
count    250.000000
mean       0.640000
std        0.480963
min        0.000000
25%        0.000000
50%        1.000000
75%        1.000000
max        1.000000
```

3. It can be inferred from above that the class level with value 1 has higher frequency compared to class level with value 0.
4. A logistic regression model was constructed and used for preliminary analysis and prediction on the dataset. The model achieved a 100% accuracy on the training dataset while only a 66.2% accuracy on the test set when evaluated on Kaggle. This clearly indicates that the model has overfitted the data.
5. Extra Tree Classifier was used for selecting significant features from the dataset and 25 features were selected. A heat map of the features was plotted and it was found that the features are not highly correlated. Extra Tree Classifier, like Random Forest,

randomizes certain decision and subsets of data to minimize overlearning from the data and overfitting.

6. It was also observed that the features are normally distributed and unimodal.

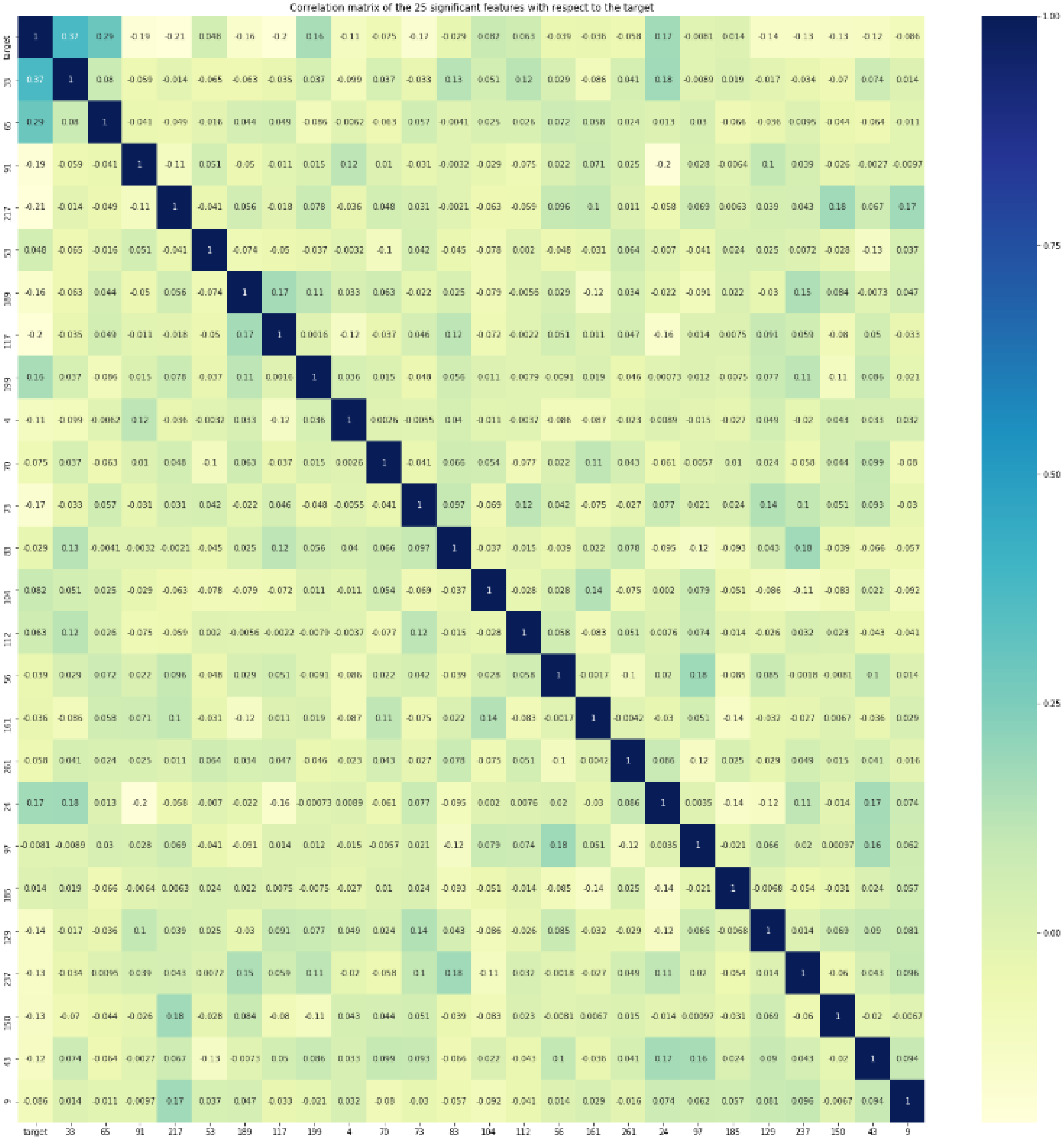


Fig 3. Heap map of the significant features selected with respect to the target vector. As can be observed from the heat map, the significant features are weakly correlated.

VI. WORK DONE AND NEXT STEPS

A simple logistic regression model was trained and tested on the original features. The model gave training accuracy of 100% but AUC-ROC [7] value of 0.662. This clearly states that a logistic model overfits if no feature extraction/selection is done. Our next step will be to come up with a strategy for feature selection and an appropriate model which should perform better than a simple logistic model.

REFERENCES

[1] https://www.kaggle.com/c/dont-overfit-ii/overview

[2] https://www.geeksforgeeks.org/dimensionality-reduction/

[3] https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

[4] https://en.wikipedia.org/wiki/Overfitting#Remedy

[5] https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114

[6] https://www.theanalysisfactor.com/the-distribution-of-independent-variables-in-regression-models/#targetText=There%20are%20NO%20assumptions%20in,continuous%20or%20discrete)%20independent%20variables.&targetText=They%20do%20not%20need%20to%20be%20normally%20distributed%20or%20continuous.

[7] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

[8] Lecture Notes