Prediction of Video Game Development Problems Based on Postmortems using Different Word Embedding Techniques

Anirudh A[1], AMAN RAJ SINGH[2], Anjali Goyal[3], Lov Kumar[4], N L Bhanu Murthy[5]
BITS Pilani Hyderabad[1,2,4,5]
AMITY University[3]
(f20180936[1],f20191483[2],lovkumar[4],bhanu[5])@hyderabad.bits-pilani.ac.in
anjaligoyal19@yahoo.in[3]

Abstract

The interactive entertainment industry is being actively involved with the development, marketing and sale of video games in the past decade. The increasing interest in video games has led to an increase in video game development techniques and methods. It has emerged as an immensely large sector, and now it has grown to be larger than the movie and music industry combined. The postmortem of a game outlines and analyzes the game's history, team goals, what went right, and what went wrong with the game. Despite its significance, there is little understanding related to the challenges encountered by the programmers. Postmortems are not properly maintained and are informally written, leading to a lack of trustworthiness. In this study, we perform a systematic analysis on different problems faced in the video game development. The need for automation and ML techniques arises because it could help game developers easily identify the exact problem from the description, and hence be able to easily find a solution. This work could also help developers in identifying frequent mistakes that could be avoided, and will provide researchers a beginning point to further consider game development in context of software engineering.

1 Introduction

The video game industry is engaged in the process of development, promotion, and selling video games. It includes several occupation disciplines and employs a huge number of individuals across the globe. The business has developed from focused markets to the mainstream in recent years. Despite being an extremely competitive market where knowledge is the principle weapon, absence of information regarding processes and techniques used in game development makes it hard to understand the game development process. Due to this, developers often find it difficult to avoid commonly occurring issues and learn from past faults.

The motive behind this work is to classify the dataset based on the quote into types of problems so that future developers could easily recognize the type of problem they are facing and find solution accordingly. However, there are 3 main challenges in this process:

- ***Word Embedding:*** The post-mortems of game development are not well structured (Washburn Jr et al., 2016). This poses an intrinsic challenge. Since the input to any machine learning (ML) model is a feature vector, it is crucial to give a numerical representation of the textual data. This challenge can be reduced by using word embedding techniques. Word embedding techniques not only give a numerical representation of the textual data, but also combine the words with similar meaning and provide a reduced set of features (Li and Yang, 2018). In this work, 7 word embedding techniques - TFIDF, Skip gram, CBOW, Word2Vec, BERT, GloVe, and FastText are applied on the text and their predictive abilities are compared.

- ***Number of Features:*** The efficiency of any ML model relies up on its features. Research (Cai et al., 2018) suggests that models where the input consisted of redundant and irrelevant features performed less efficiently. Since the data set consists of a huge number of features, this poses an intrinsic challenge. To overcome this, 3 feature selection techniques have been used to select the relevant and crucial features: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Analysis of variance (ANOVA).

- ***Class Imbalance:*** A balanced dataset (Jun-

Proceedings of the 18th International Conference on Natural Language Processing, pages 465–473
Silchar, India. December 16 - 19, 2021.

somboon and Phienthrakul, 2017) is one that contains an equal or almost equal number of samples from all the dependent variables. The last challenge in building the predictive model is that the data suffers from class imbalance problem. Hence, Synthetic Minority Oversampling Technique (SMOTE) has been used to balance the data.

To overcome the 3 challenges, a technical analysis and comparative study between the performance of 7 word embedding, 3 feature selection, and 5 classification techniques have been conducted. 5 classification techniques namely, K-Nearest Neighbours (KNN), Support Vector Classifier (SVC), Naive Bayes Classifier (NBC), Decision Tree (DT) and Random Forest (RF) have been used. Finally, the performance of the word embedding and classification techniques are compared. The results obtained from the original data is also compared to the results obtained from SMOTE data. To compare the results, accuracy, F-measure, and area under the curve (AUC) are used. Box plots are drawn based on AUC values since accuracy is not a good measure when the data suffers from class imbalance. Finally, rank sum test and friedman's test are used to test the hypotheses.

2 RELATED WORK

This section details about studies available in the broad domain of game development. (Politowski et al., 2020) prepared a grounded dataset obtained from post-mortems of video games which details about various software engineering problems during development. An iterative method has been used to create the dataset. 1035 problems were extracted from more than 200 post-mortems spanning over 20 years (1998-2018). The problems are divided into 3 problem groups (production, business, management) which are further divided into 20 different types. This work utilizes the above stated dataset to understand issues encountered by developers during the process of video-game development and further we foster a model for distinguishing the problem group based on description.

Game industry problems: An extensive analysis of the gray literature (Politowski et al., 2021) tries to analyse and develop a state of the problems of the gaming industry, their evolution and root causes which would help researchers and practitioners to work towards addressing and solving these problems. It was observed that the industry suffers from similar proportions of management and production related problems. Over the years as the industry became more mainstream, management related problems decreased only to give space to business related problems. Technical and design related problems have also decreased over the years. Team related problems increased over the last decade, and marketing problems had the biggest increase over the past 23 years. Finally, it was concluded that people (and not technology) were the root cause of most problems.

Callele et al. (Callele et al., 2005) analysed the various factors which led to the success or failure of a video game. Using the Game Developer Magazine, they analysed 50 post-mortems and investigated how requirements engineering could be applied to game development. "What went right" and "What went wrong" aspects were grouped into 5 categories: (1) pre-production issues (2) internal, and management related problems (3) external problems (4) technological problems (5) scheduler related problems. Finally, it was concluded that the transition from preproduction to production plays a crucial role in deciding the fate (success/failure) of the developed video game.

20 post-mortems were taken from the Gamasutra Website and were analysed by Petrillo et al. (Petrillo et al., 2009) The most common game development problems were identified and they were compared with traditional software-engineering problems. It was concluded that (1) management (and not technical) related issues contribute most to the video-game development problems (2) problems faced in video-game development and traditional software development are very similar and, (3) the most common problems are related to scope, feature creep, and cutting features.

Washburn et al. (Washburn Jr et al., 2016) analysed 155 game development post-mortems for what went right and wrong. Various characteristics of game development have been identified, linked with positive and negative experiences and a set of best practices, pitfalls for game development have been distilled. Design aspects cover all situations and decisions that were made that are external to the direct team and development process. Production issues relate to scheduling and work prioritizing issues. Other aspects include art, programming and testing issues.

3 Study Design

This section enlists the information about different design plans utilized in this work.

3.1 Experimental Dataset

This work uses Video Game Development problems data set which was collected from MSR 2020 conference datasets (Politowski et al., 2020). The dataset was created using iterative method, where more than 200 post-mortems for different video games were studied and around 1035 problems related to software engineering were extracted. The problems were divided into 3 groups – production, management, and business (Politowski et al., 2021). Production problems can be classified as problems based on documentation, prototyping, technical, testing, tools, bugs, and design. Business problems could be due to marketing or monetization, while management problems could be classified as problems based on communication, crunch time, delays, team, budget, planning, security, scope, cutting features, feature creep, and multiple projects.

3.2 Word Embedding

The representation of words for text analysis, in the form of a real-valued vector is called word embedding. The feature vectors encode meaning of the word such that words with similar meaning are closer in vector space. This reduces the overall feature space. Primarily, there are 2 word embedding methods: frequency-based, and neural network-based. In this work, 7 word embedding techniques have been applied to represent the words as a vector in n-dimensional vector space. The data has been cleaned by removing stop-words, bad symbols, spaces, etc. Further, predictive power of word embedding techniques have been compared.

3.3 SMOTE

The considered dataset suffers from the problem of class imbalance. Out of the four categories, the maximum class has around 430 data points while the minority class has less than 100 data points. Since ML algorithms increase accuracy by reducing the error, the class distribution is not considered. This problem is also prevalent in various domains such as fraud or anomaly detection, face identification, etc. Conventional ML algorithms such as logistic regression, DT, etc. possess bias towards majority class (Hoens and Chawla, 2013). Hence, dataset is balanced using SMOTE technique (Fernández et al., 2018)(Chawla, 2009). SMOTE balances the class distribution by replicating minority class instances.

3.4 Feature Selection

This work utilizes 3 feature selection techniques: ANOVA, PCA, and LDA for eliminating irrelevant features. The predictive power of the classifiers learnt using selected features is compared with the predictive power of the classifiers learnt using all features using AUC, F-measure, and accuracy. Further, rank sum test has been applied.

- ***ANOVA*** is a collection of statistical models for analysing differences among means (Sarstedt and Mooi, 2019)(St et al., 1989). The one-way classification follows completely random design (CRD), while two-way classification follows random block design (RBD). Overall, ANOVA possess no assumptions. However, CRD assumes independence, normality and homogeneity of variances of the residuals while RBD assumes homogeneity of variances of residuals. ANOVA partitions total sum of squares (SS) into components related to the effects used in model. For instance, model for a simplified ANOVA with one type of treatment at different levels would have $SS_{Total} = SS_{Error} + SS_{Treatments}$. For comparing factors of total deviation, below formula for F-test is used:

$$F = \frac{Variance\ \ between\ \ treatments}{Variance\ \ within\ \ treatments} \quad (1)$$

- ***PCA*** is an unsupervised dimensionality-reduction technique to transform large number of features into a smaller set which comprises similar information as contained by large number of features. Each instance is projected onto only principal components to reduce dimensionality while preserving data variation. If variance is high, it is easier to find patterns in the data set and so, we choose the ones with high variance as the important features.

- ***LDA*** is a supervised learning technique for dimensionality reduction where classes and their dependencies are also considered (Martinez and Kak, 2001)(Yu and Yang, 2001). Unlike PCA which considers maximum variance alone, LDA considers within class and between class variance also. The objective of

LDA is to extend components from higher dimensional space onto a lower dimensional space to keep away from the curse of dimensionality and furthermore decrease resources and dimensional expenses. Reducing the dimensions shrinks and concludes the dimensions which helps in better understanding of the data. LDA measures data from all features to make a new axis that limits variance and maximizes class distance.

3.5 Classification Techniques

The performance of various word-embedding, feature selection and SMOTE is evaluated with 5 ML classifiers: KNN, SVC, NBC, DT, & RF.

4 RESEARCH METHODOLOGY

The motive behind this work is to do a technical analysis and comparison between performance of 7 word embedding and 5 classification techniques on game development problems. The algorithm tries to classify the dataset based on the quote into types of problem so that future developers could easily recognize the type of problem they are facing and find solution accordingly. Firstly, 7 word embedding techniques are applied on text available in the dataset to obtain feature vectors. Next, data imbalance problem was dealt with using SMOTE technique which adds more minority test cases and balances data. Dimensionality reduction and feature selection was done using PCA, LDA and ANOVA. Finally, 5 classification techniques were used to predict class of the test data using embedded vectors. The performance of various word embedding and classification techniques are compared. The results obtained from original data are also compared to that obtained from SMOTE data using accuracy, F-measure and AUC. Box plots are drawn based on the AUC values since accuracy is not a good measure when the data suffers from class imbalance (Bekkar et al., 2013). Rank sum test and Friedman's test are used to test the hypotheses. Framework of the proposed work is depicted in Figure 1.

5 Empirical Results and Analysis

In this work, 7 word embedding, 1 sampling, 3 feature selection, and 5 ML classifiers were applied to develop models which predict the group of game development problem. Each word-embedding is applied on the chosen dataset and its effectiveness is evaluated using classifiers. The predicted values for each of the word embedding and classification techniques have been tabulated in Table 1. AUC, accuracy, and F-Measure values are calculated. To compare the results, however, only AUC values are used. This is because accuracy is not a good measure when there is class imbalance and AUC uses probability measures. Table 1 denotes AUC values corresponding to the original and SMOTE sampled data. Figure 2 shows the bar graph of AUC score for models trained on original data and balanced data for different sets of features. The models are validated using 5-fold cross validation (CV). AF denotes the AUC values corresponding to all features while ANOVA, PCA, LDA correspond to those got after feature selection. From Table 1 and Figure 2, we can infer following:

- High values of AUC confirm that developed models can predict different video game development problems based on the data.

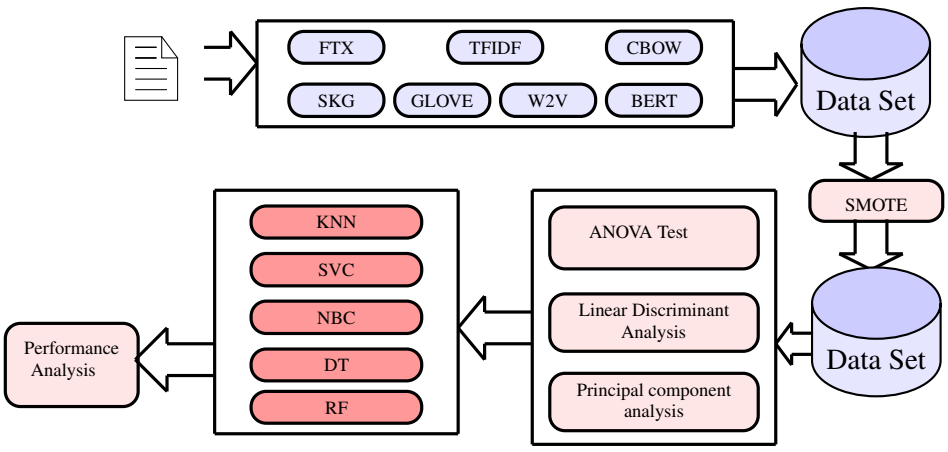
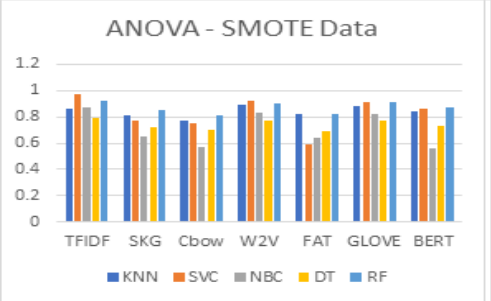


Figure 1: Research Framework

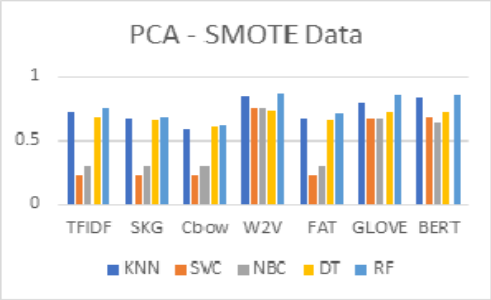
Table 1: AUC values

		Original Data					SMOTE Data				
		KNN	SVC	NBC	DT	RF	KNN	SVC	NBC	DT	RF
AF	TFIDF	0.54	0.82	0.53	0.62	0.70	0.83	0.99	0.96	0.80	0.92
	SKG	0.58	0.75	0.63	0.53	0.61	0.80	0.77	0.64	0.71	0.83
	Cbow	0.55	0.74	0.55	0.51	0.59	0.76	0.77	0.54	0.69	0.79
	W2V	0.79	0.82	0.80	0.59	0.70	0.89	0.93	0.83	0.74	0.90
	FAT	0.55	0.61	0.59	0.51	0.56	0.81	0.59	0.61	0.69	0.81
	GLOVE	0.78	0.82	0.80	0.59	0.74	0.89	0.91	0.82	0.76	0.91
	BERT	0.60	0.71	0.53	0.54	0.57	0.84	0.88	0.56	0.74	0.88
ANOVA	TFIDF	0.60	0.90	0.73	0.61	0.77	0.86	0.97	0.87	0.79	0.92
	SKG	0.57	0.75	0.63	0.54	0.63	0.81	0.77	0.65	0.72	0.85
	Cbow	0.54	0.74	0.55	0.53	0.59	0.77	0.75	0.57	0.70	0.81
	W2V	0.79	0.83	0.81	0.57	0.72	0.89	0.93	0.83	0.77	0.90
	FAT	0.55	0.60	0.62	0.51	0.56	0.83	0.59	0.64	0.69	0.83
	GLOVE	0.78	0.82	0.80	0.60	0.70	0.89	0.91	0.82	0.77	0.91
	BERT	0.59	0.71	0.54	0.53	0.57	0.84	0.86	0.56	0.73	0.87
PCA	TFIDF	0.59	0.44	0.50	0.52	0.57	0.73	0.22	0.30	0.69	0.76
	SKG	0.52	0.43	0.50	0.52	0.51	0.67	0.22	0.29	0.66	0.69
	Cbow	0.54	0.43	0.50	0.53	0.54	0.60	0.22	0.29	0.61	0.62
	W2V	0.68	0.77	0.77	0.60	0.70	0.85	0.76	0.76	0.74	0.87
	FAT	0.50	0.47	0.50	0.52	0.52	0.68	0.22	0.29	0.66	0.71
	GLOVE	0.63	0.73	0.72	0.57	0.66	0.80	0.68	0.67	0.73	0.85
	BERT	0.58	0.66	0.60	0.55	0.58	0.84	0.68	0.64	0.72	0.86
LDA	TFIDF	0.93	0.88	0.92	0.87	0.95	0.97	0.86	0.92	0.92	0.97
	SKG	0.94	0.97	0.97	0.87	0.95	0.97	0.97	0.97	0.91	0.98
	Cbow	0.94	0.96	0.97	0.84	0.94	0.96	0.97	0.97	0.89	0.97
	W2V	0.94	0.97	0.97	0.85	0.95	0.97	0.97	0.97	0.90	0.97
	FAT	0.78	0.84	0.84	0.66	0.78	0.87	0.84	0.84	0.78	0.89
	GLOVE	0.94	0.97	0.97	0.84	0.94	0.97	0.97	0.97	0.90	0.97
	BERT	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.99	1.00

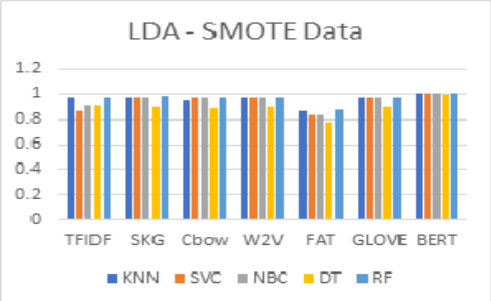
- Word2Vec embedding gives the best results.
- Results of FastText are poorer than others.
- Model trained using RF has better predictions.
- Model trained on SMOTE data has better AUC score than original data.



(2.1) AUC: ANOVA on SMOTE



(2.2) AUC: PCA on SMOTE



(2.3) AUC: LDA on SMOTE

Figure 2: AUC vlaue

6 Comparative Analysis

In this section, we compare the performance of models built using 7 word embedding, SMOTE, 3 feature selection, & 5 classification techniques. Descriptive statistics, box-plots, & significant tests have been used to compare developed models.

6.1 Word Embedding

In this work, 7 word embedding techniques (TFIDF, Skipgram, CBOW, Word2vec, FastText, GloVe and BERT) have been applied to represent words as vectors in n-dimensional vector space. The data has been cleaned before these techniques were applied, i.e., stop-words, bad symbols, spaces, etc. have been removed. These techniques not only give a numerical representation of textual data, but also encode their meaning such that words which are similar in meaning are closer in vector space. Moreover, as compared to a large set of vocabulary, a small number of features is obtained. The predictive ability of developed models using word embeddings are computed with the help of AUC score, F-Measure, and accuracy value. However, only AUC scores are considered for comparison since the data suffers from class imbalance. The AUC values are compared using descriptive statistics, box-plots, and significant tests.

6.1.1 Box-Plot: Word Embedding

Figure 3 provides descriptive statistics and performance values (measured using AUC) of 7 word embedding techniques in terms of a box-plot. From Figure 3, it is clear that models developed using Word2Vec, TFIDF, GloVe better predict group of game development problem. As compared to other techniques, models developed using CBOW, FastText, Skipgram have a low predictive ability. This is evident from the fact that mean AUC scores of CBOW, Skipgram & FastText are 0.66, 0.70 & 0.64 respectively while mean AUC scores of TFIDF,

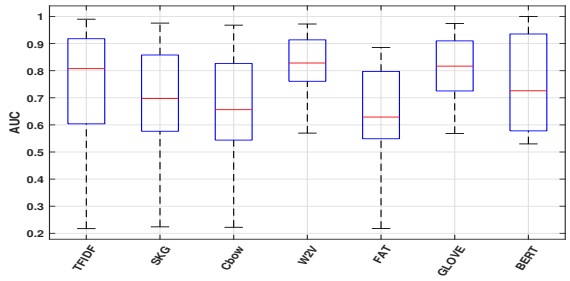


Figure 3: AUC Scores for Word Embedding Techniques

Word2vec & GloVe are 0.80, 0.83 & 0.82 respectively.

6.1.2 Significant Tests: Word Embedding

In this study, Friedman's test and rank-sum test are applied on the AUC scores to statistically compare the predictive ability and performance of the models built using 7 word embedding techniques. To check if these models have a significant improvement on the predictive ability or not, the following hypothesis has been formed:

- ***Null Hypothesis:*** The performance of the models do not depend on features extracted from word embedding techniques.
- ***Alternate Hypothesis:*** The performance of the models depend on features extracted from word embedding techniques.

To test the hypothesis, Friedman's test and rank sum test are used with a significance level of 0.05 i.e., null hypothesis is accepted if p≥0.05. For the purpose of simplicity a two-number representation of results has been used, i.e., 0 if null hypothesis is accepted (models are significantly same) and 1 if hypothesis is rejected (models are significantly different). From Table 2, it can be seen that Skipgram and CBOW give significantly different results as compared to Word2Vec, GloVe. Similarly, FastText gives significantly worse results as compared to TFIDF, GloVe, BERT.

Further, since models prove to give significantly different results, Friedman's mean rank test is also applied on AUC values to rank 7 word embedding models. A model with a lower mean rank value performs better than the one with a higher mean rank value. Hence, from Table 4 we can conclude that w2v gives the best results (followed by GloVe) and that its performance is significantly better than Skipgram, CBOW and FastText.

Table 2: Rank-Sum Test: Word Embedding

	Rank-Sum							Friedman's
	TFIDF	**SKG**	**CBOW**	**W2V**	**FAT**	**GLOVE**	**BERT**	**Mean-Rank**
TFIDF	0	0	0	0	1	0	0	3.32
SKG	0	0	0	1	0	1	0	4.30
CBOW	0	0	0	1	0	1	0	5.82
W2V	0	1	1	0	1	0	0	2.27
FAT	1	0	0	1	0	1	1	6.27
GLOVE	0	1	1	0	1	0	0	2.625
BERT	0	0	0	0	1	0	0	3.37

6.2 SMOTE

In this paper, it has been proposed to apply SMOTE in order to get balanced data and account for class imbalance. In this section, the predictive ability of prediction models using original data and SMOTE sampled data are compared using descriptive statistics, boxplot diagram, and significant tests.

6.2.1 Box Plots: Original and SMOTE data

The AUC values of the models developed using original data and SMOTE sampled data have been compared using box-plot diagrams as shown in Figure 4 and descriptive statistics. The information in Figure 4 shows that the models developed using SMOTE sampled data achieved 0.82 mean AUC score while that of original data achieved 0.64 mean AUC value. Hence, it can be concluded that SMOTE data sampling technique plays a crucial role in enhancing the model's ability to predict the group of game development problem.

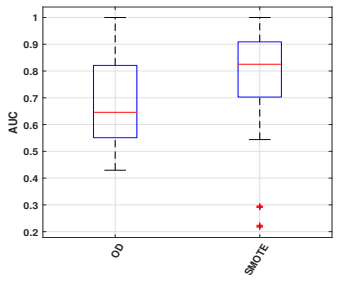


Figure 4: AUC Scores for Original data and SMOTE data

6.2.2 Significant Tests: Original and SMOTE data

In this study, Friedman's test and rank sum test are applied on the AUC scores to statistically compare the predictive ability of the models built using the original data and SMOTE sampled data. The objective is to check if the models developed using SMOTE has a significant improvement on the predictive ability or not, and for this, the following hypothesis has been formed:

Table 3: Significant Tests: Original and SMOTE data

Rank-Sum			Friedman's
	OD	SMOTE	Mean-Rank
OD	0	1	1.871
SMOTE	1	0	1.128

- ***Null Hypothesis (H_0)***: The models developed using SMOTE sampled data do not have a significant impact on the predictive ability of classification model.

- ***Alternate Hypothesis (H_a)***:The models developed using SMOTE sampled data have a significant impact on the predictive ability of classification model.

To test above hypothesis, p-value is used. Considering a significance level of 0.05 level (95% confidence interval), null hypothesis is accepted if the p-value$\geq$0.05. From Table 3, it can be seen that the models built from the original data and SMOTE sampled data are significantly different. From Table 3, we can also conclude that the model developed using SMOTE sampled data performs significantly better and that accounting for class imbalance is crucial.

6.3 Feature Selection

In this work, 3 feature selection techniques were used to remove unnecessary features and for dimensionality reduction. The 3 feature selection techniques – ANOVA (Figure 22.1), PCA (Figure 22.2), and LDA (Figure 22.3) have been applied to find best combination of relevent features. The predictive ability of the developed models using the 3 feature selection techniques are evaluated using AUC scores. They are compared using descriptive statistics, boxplot diagram, and significant tests.

6.3.1 Box Plots: Different sets of Features

Figure 5 provides the descriptive statistics and performance values (measured using AUC) of the 3 feature selection techniques in terms of a box-plot. From Figure 5, it is evident that the models developed using LDA best predict the group of game development problem. The models developed using ANOVA, PCA have a relatively low predictive ability as compared to LDA. This is evident from the fact that the mean AUC scores ANOVA, PCA, LDA are 0.75, 0.6, 0.97 respectively.

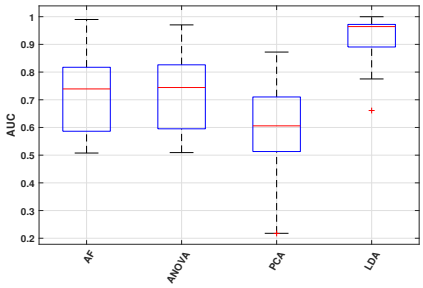


Figure 5: Feature Selection

Table 4: Significant Tests: Different sets of Features

Rank-Sum					Friedman's
	AF	ANOVA	PCA	LDA	Mean-Rank
AF	0	0	1	1	2.757
ANOVA	0	0	1	1	2.428
PCA	1	1	0	1	3.757
LDA	1	1	1	0	1.057

6.3.2 Significant Tests: Different sets of Features

In this study, Friedman's test and rank sum test are applied on the AUC scores to statistically compare the predictive ability of the models developed from 3 feature selection techniques. To test the hypothesis, p-value is used at significance level of 0.05 level (95% confidence interval), null hypothesis is accepted if the p-value $>$ 0.05. For the purpose of simplicity, a two-number representation for the results has been used, i.e., 0 if the null hypothesis is accepted (models are significantly same) and 1 if the hypothesis is rejected (models are significantly different). From Table 4, it can be seen that the models developed using PCA, LDA are significantly different from the model developed using all features. Similarly, models developed using LDA is significantly different from all other models.

Further since models prove to give significantly different results, Friedman's mean rank test is performed on AUC values to rank the models developed using feature selection techniques. A model with a lower mean rank value performs better than one with a higher mean rank value. Hence, from Table 4 we can conclude that model developed using LDA gives best results while model developed using PCA gives worst prediction results. Also, it can be seen that models built using ANOVA, all features are not significantly different from one another. These results also verify the conclusion made from the box plot in Figure 5.

6.4 Classification Techniques

In this study, KNN, SVC, NBC, DT, RF have been used to classify the game development problem into 3 groups (production, management, and business-related problems). 5-fold CV has been used to train the prediction models. In this section, the predictive ability of the models developed using 5 classifiers are computed using AUC scores. The AUC values are compared using descriptive statistics, box-plots, & significant tests.

6.4.1 Box Plots: Classification Techniques

Figure 6 provides descriptive statistics and performance values (measured using AUC) of 5 classification techniques in terms of a box-plot. From Figure 6, it is clear that models developed using KNN, SVC, RF better predict group of game development problem. Compared to models built using KNN, SVC, and RF, models developed using NBC, DT have a low predictive ability. This is evident from the fact that mean AUC scores of KNN, SVC & RF are 0.80, 0.80 & 0.81 while mean AUC scores of NBC & DT are 0.70 & 0.69 respectively.

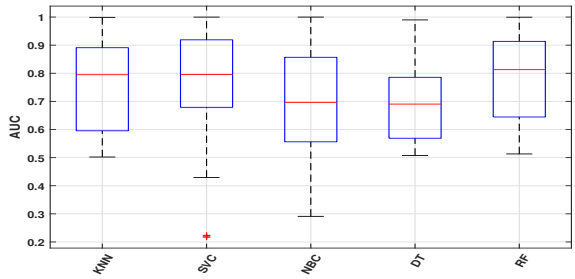


Figure 6: Classification Techniques

6.4.2 Significant Tests: Classification Techniques

In this study, Friedman's test and rank sum test are applied on AUC scores to statistically compare the predictive ability of models developed using 5 classifiers. Table 5 show results of Friedman's test and rank sum test for different techniques. For the purpose of simplicity, two-number representation for the results is used, i.e., 0 if null hypothesis is accepted (models are significantly same) and 1 if hypothesis is rejected (models are significantly different). From Table 5, it can be seen that model developed using DT is significantly different from models developed using KNN, SVC, RF while it does not differ significantly from NBC in terms of predictive ability. Further since the models prove to give significantly different results, Friedman's mean rank test is also applied on AUC values to rank 5 models. A lower value of mean rank indicates a better performance of the model. Hence, from Table 5, we can conclude that RF classifier gives the best results (followed by SVC, KNN) and its performance is significantly better than model built using DT.

Table 5: Significant Tests: Classification Techniques

	Rank-Sum					Friedman's
	KNN	**SVC**	**NBC**	**DT**	**RF**	**Mean-Rank**
KNN	0	0	0	1	0	3.000
SVC	0	0	0	1	0	2.375
NBC	0	0	0	0	0	3.107
DT	1	1	0	0	1	4.303
RF	0	0	0	1	0	2.214

7 Conclusion

A postmortem is a summarization procedure used to analyse the various positive and negative aspects of the game development project. It aids developers in drawing meaningful conclusions and helps them learn from past successes and failures. However, given the various responsibilities of a video game developer, it is not surprising that they hardly take time to conduct and prepare project postmortems. Moreover, the lack of formal structure leads to a lack of trust worthiness. In this work, a data set of the different problems in game development has been taken and studied. 7 word embedding techniques have been applied on the data set and it can be observed that Word2Vec gives the best results and these results are significantly different from Skipgram, CBOW and FastText. As far as the classification techniques are concerned, KNN, SVC, and RF produce significantly better results. RF gives the best value in Friedman's rank sum test. However, KNN may also be considered since it takes the lowest computation time amongst KNN, SVC, RF whilst still producing similar results. LDA is the best suitable feature selection technique here, followed by ANOVA. Finally, it is important to note that the data obtained after SMOTE gives significantly better yield than the original data, and hence, accounting for class imbalance is crucial. This work, thus provides a way to classify the dataset based on the quote into types of problems. This could help future developers easily recognize the type of problem they are facing and find suitable solutions.

References

Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. 2013. Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl*, 3(10).

Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. 2018. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79.

David Callele, Eric Neufeld, and Kevin Schneider. 2005. Requirements engineering and the creative process in the video game industry. In *13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 240–250. IEEE.

Nitesh V Chawla. 2009. Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, pages 875–886.

Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. 2018. *Learning from imbalanced data sets*, volume 10. Springer.

T Ryan Hoens and Nitesh V Chawla. 2013. Imbalanced datasets: from sampling to classifiers. *Imbalanced learning: Foundations, algorithms, and applications*, pages 43–59.

Nutthaporn Junsomboon and Tanasanee Phienthrakul. 2017. Combining over-sampling and under-sampling techniques for imbalance dataset. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 243–247.

Yang Li and Tao Yang. 2018. Word embedding for understanding natural language: a survey. In *Guide to big data applications*, pages 83–104. Springer.

Aleix M Martinez and Avinash C Kak. 2001. Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233.

Fábio Petrillo, Marcelo Pimenta, Francisco Trindade, and Carlos Dietrich. 2009. What went wrong? a survey of problems in game development. *Computers in Entertainment (CIE)*, 7(1):1–22.

Cristiano Politowski, Fabio Petrillo, Gabriel C Ullmann, and Yann-Gaël Guéhéneuc. 2021. Game industry problems: An extensive analysis of the gray literature. *Information and Software Technology*, 134:106538.

Cristiano Politowski, Fabio Petrillo, Gabriel Cavalheiro Ullmann, Josias de Andrade Werly, and Yann-Gaël Guéhéneuc. 2020. Dataset of video game development problems. In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 553–557.

Marko Sarstedt and Erik Mooi. 2019. Hypothesis testing and anova. In *A Concise Guide to Market Research*, pages 151–208. Springer.

Lars St, Svante Wold, et al. 1989. Analysis of variance (anova). *Chemometrics and intelligent laboratory systems*, 6(4):259–272.

Michael Washburn Jr, Pavithra Sathiyanarayanan, Meiyappan Nagappan, Thomas Zimmermann, and Christian Bird. 2016. What went right and what went wrong: an analysis of 155 postmortems from game development. In *Proceedings of the 38th International Conference on Software Engineering Companion*, pages 280–289.

Hua Yu and Jie Yang. 2001. A direct lda algorithm for high-dimensional data—with application to face recognition. *Pattern recognition*, 34(10):2067–2070.