A

Major Project

On

EVALUATION OF EDUCATION APPS WITH APP STORE DATA

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By

Pokala Rakesh(187R1A0544)

Veluguri Anand Prakash(187R1A0559)

Mukul Chaubey(187R1A0524)

Under the Guidance of

B. P. DEEPAK KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CMR TECHNICAL CAMPUS
UGC AUTONOMOUS
(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956
Kandlakoya (V), Medchal Road, Hyderabad-501401.
2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**EVALUATION OF EDUCATION APPS WITH APP STORE DATA**" being submitted by **P. RAKESH (187R1A0544), V. ANAND PRAKASH (187R1A0559) & MUKUL CHAUBEY (187R1A0524)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. B. P. Deepak Kumar
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
Hod

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mr. B. P. Deepak Kumar,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work.The blessing,help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. A. Uday Kiran, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Mrs. G. LathaMr, A. Kiran Kumar,** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

POKALA RAKESH (187R1A0544)
VELGURI ANAND PRAKASH (187R1A0559)
MUKUL CHAUBEY (187R1A0524)

ABSTRACT

In recent days, due to rise in the number of applications that are available in application stores and the complexity of the features they provide, choosing a good app for learning has become challenging for teachers,students and parents. To assist people in selecting a good application there are several evaluation frameworks which are proposed in literature. One of the most well established framework is the iPAC framework which highlights the learner's experience in terms of personalization,authenticity and collaboration. But most of the literature methods are non-automatic and requires human effort hence we are proposing a system which uses natural language processing and machine learning techniques. These techniques uses data that is collected from the apps reviews and from the description provided by the application owner which are available on the application store. We evaluate the app based on the positive and negative reviews that are provided by the users and validating them by the positive and negative keywords. We used several classification algorithms to determine a single algorithm which can give us good accuracy results.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

LIST OF TABLES

TABLE OF CONTENTS

1.INTRODUCTION

2. SYSTEM ANALYSIS

3. ARCHITECTURE

4. IMPLEMENTATION

5.SCREEN SHOTS

6.TESTING

7.CONCLUSION

8.BIBILOGRAPHY

9.PAPER PUBLICATION

10.CERTIFICATES

1. INTRODUCTION

1.1 PROJECT SCOPE

Over the past decade, research has shown that mobile learning is a promising approach to improve learning effectiveness and experience. For instance, various studies have examined and demonstrated the positive effect of using mobile technology and mobile apps on the students' learning outcomes.But our concern is that there are many educational applications present in the play store which could give a lot of confusion to users while selecting a application. In order make the task of searching easy we can evaluate them using machine learning.

1.2 PROJECT PURPOSE

The educational mobile apps offer innovative opportunities for teachers to improve students' learning. Over the past decade, researchers have investigated the effectiveness of apps in various education domains and proposed new pedagogical frameworks for mobile learning. However, with the vast, continuously increasing number of available apps, choosing "the right app" is becoming more and more difficult and time-consuming for teachers and students . One particular challenge is to efficiently select an app that appropriately supports the desired learning activities, assessment strategies, and pedagogical preferences.

1.3 PROJECT FEATURES

The main features of this project are that the designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

Currently, The educational mobile apps offer innovative opportunities for teachers to improve students’ learning. Over the past decade, researchers have investigated the effectiveness of apps in various education domains and proposed new pedagogical frameworks for mobile learning. However, with the vast, continuously increasing number of available apps, choosing “the right app” is becoming more and more difficult and time-consuming for teachers and students . One particular challenge is to efficiently select an app that appropriately supports the desired learning activities, assessment strategies, and pedagogical preferences.

2.2 EXISTING SYSTEM

The “Evaluation of educational apps with app store data” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is effective manner.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Time consuming.
- Tedious process.

To avoid all these limitations and make the working more accurately the system needs to be implemented efficiently.

2.3 PROPOSED SYSTEM

In this proposed work we are implementing machine learning algorithm for searching a good application based on the reviews and description for better understanding of the application. No formal knowledge is needed for the user to use this system. Thus by this all it provides it is user-friendly. Evaluation of educational apps as described above, can lead to error free,reliable. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement.The system requires very low system resources and the system will work in almost all configurations. It got following features

- It is faster and effective than the previous existing system.
- Better service.
- User friendliness and interactive.
- Minimum time required.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out.This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for theusers?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system.The following are some hardware requirements.

- Processor Intel Dual Core@ CPU 2.90GHz.
- Hard disk 20GB and Above.
- RAM 2GB and Above.
- Monitor 5 inches or above.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system Windows 7,8,10
- Language Python
- Database MySQL 5.0

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for Evaluation of Education Apps with Apps Store Data

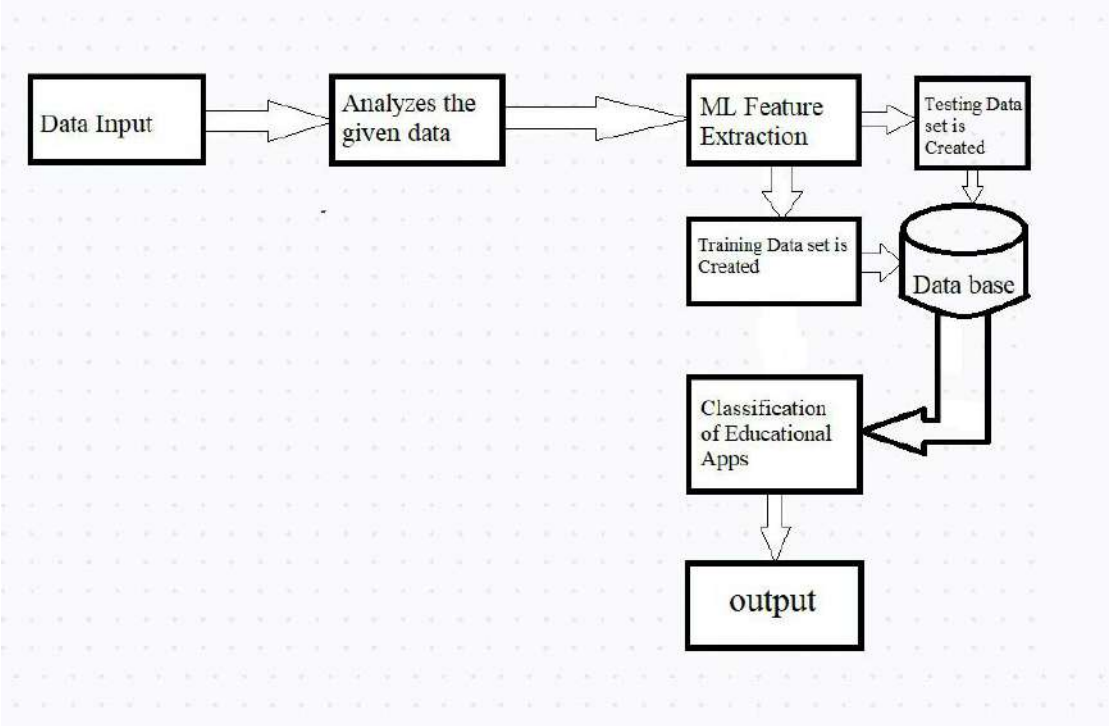


Figure 3.1: Project Architecture of evaluation of education apps with app store data

3.2 DESCRIPTION

Data Input: The data that is necessary for the purpose of our study is given.

Analyzing Data : The given data is preprocessed and is sent to the machine learning algorithm.

ML Feature Extraction: Ml feature extraction is performed create training data and test

data after creating the training data the algorithm is trained on that data.

Database: Database is used to store the data and to get the when required.

Classification: Classification of the applications based on the reviews will be done and the application with most positive reviews is found by using ml algorithm.

Output: The output is shown on the output pane.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the algorithm.

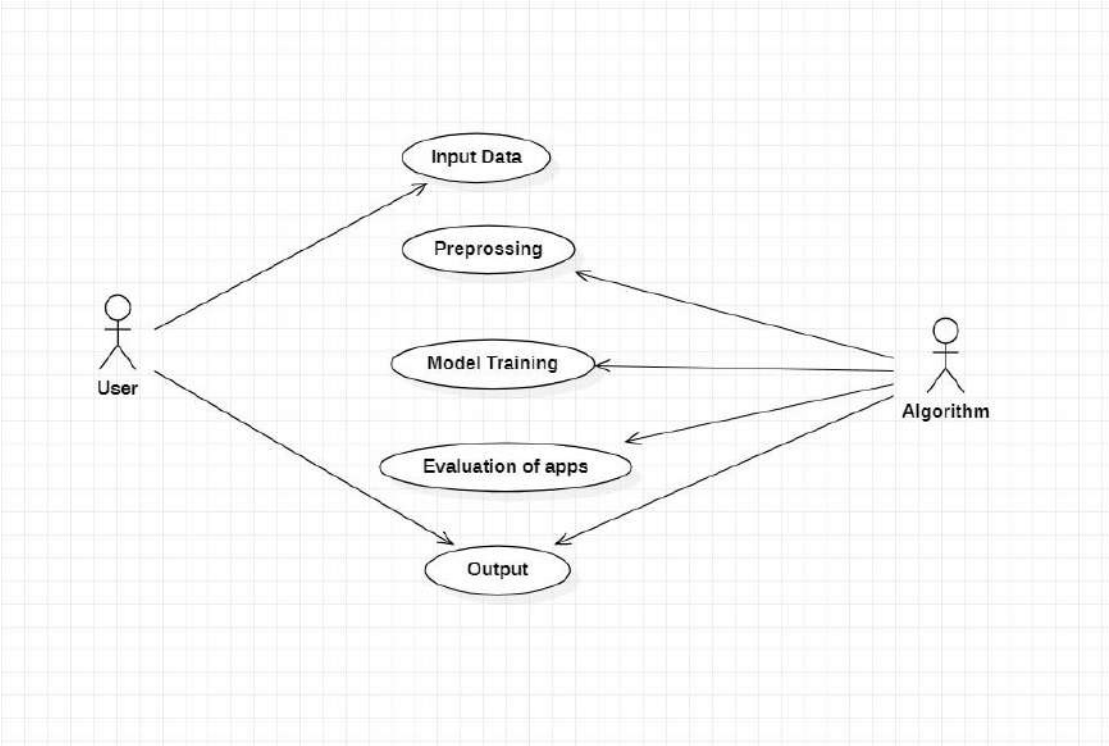


Figure 3.2: Use Case Diagram of evaluation of education apps with app store data

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

Figure 3.3: Class Diagram of evaluation of education apps with app store data

3.5 SEQUENCE DIAGRAM

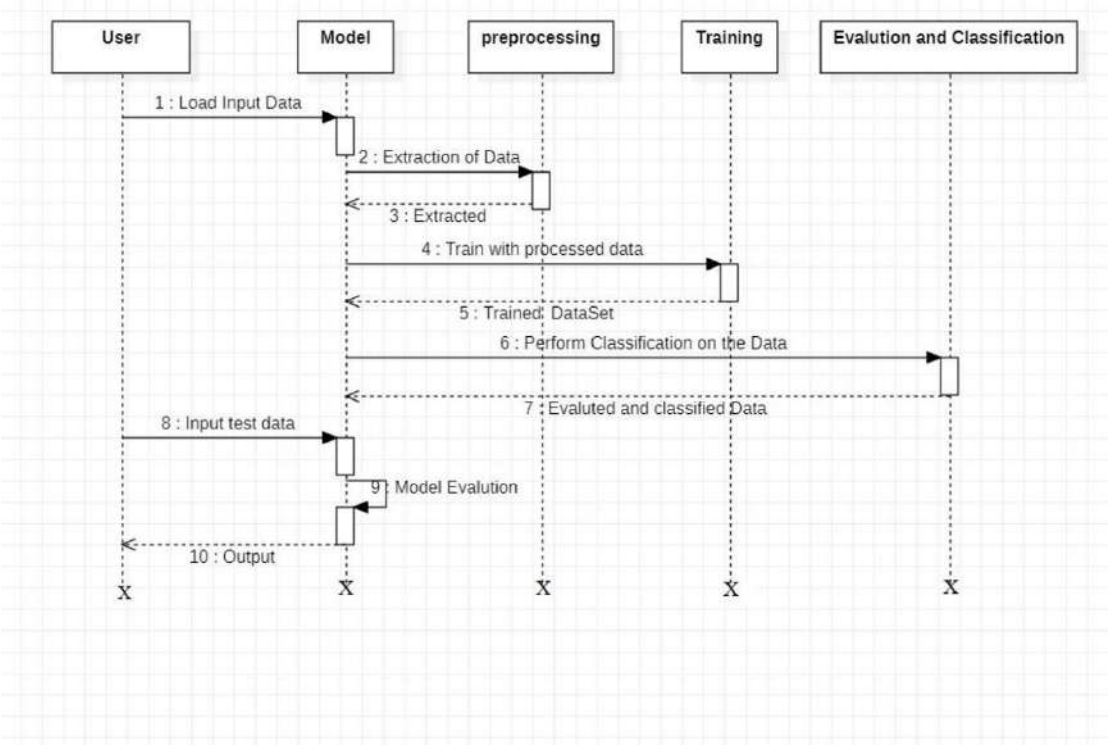


Figure 3.4: Sequence Diagram of evaluation of education apps with app store data

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

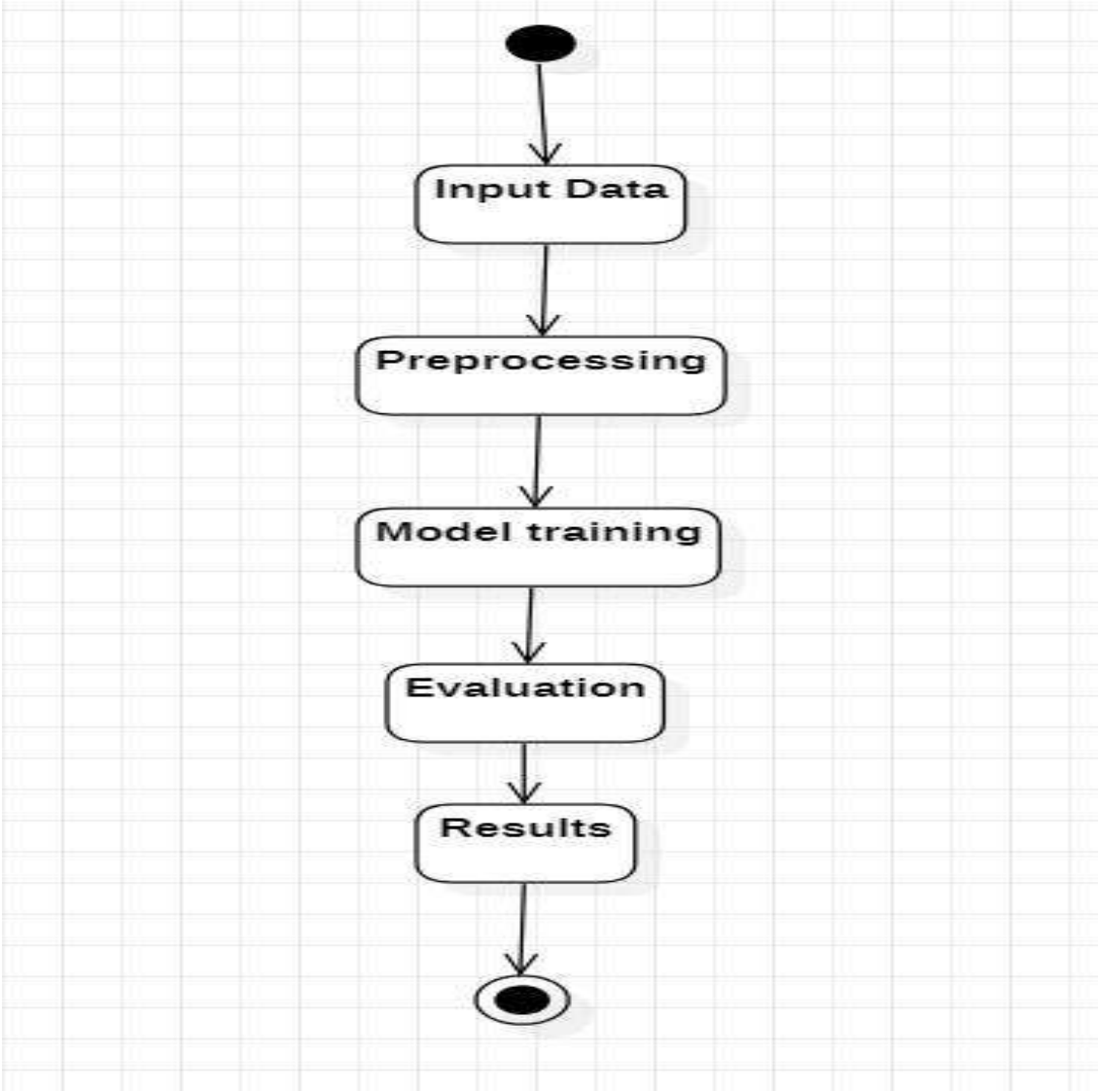


Figure 3.5: Activity Diagram of evaluation of education apps with app store data

4. IMPLEMENTATION

4.1 SAMPLE CODE

Importing Libraries

```
import pandas as pd
import numpy as np
from wordcloud import WordCloud, STOPWORDS
import plotly.graph_objs as go
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
from plotly.offline import init_notebook_mode, iplot
import plotly.offline as py
py.init_notebook_mode(connected=True)
import warnings
warnings.filterwarnings('ignore')
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer, PorterStemmer
from nltk.corpus import stopwords
from nltk.stem.lancaster import LancasterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from textblob import TextBlob
from wordcloud import WordCloud, STOPWORDS
```

Importing data sets

```
Data = pd.read_csv("app_info.csv")
data.head()
```

Data Description

```
data.info()
```

```
data.select_dtypes(include = ['object']).columns.values


data.select_dtypes(include = ['int64', 'float64']).columns.values
data.genreId.unique()
data.androidVersion.unique()
```

Text Preprocessing

```
import nltk
import string
import re
df['content']= df['content'].str.lower()
df.head()
df['content'] = df['content'].astype(str)
df.info()# remove all numbers with letters attached to them
alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)
# '[%s]' % re.escape(string.punctuation),' ' - replace punctuation with white space
# .lower() - convert all strings to lowercase
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())
# Remove all '\n' in the string and replace it with a space
remove_n = lambda x: re.sub("\n", " ", x)
# Remove all non-ascii characters
remove_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)
# Apply all the lambda functions wrote previously through .map on the comments column
 df['content'] =
 df['content'].map(alphanumeric).map(punc_lower).map(remove_n).map(remove_non_ascii)
   text = cleaner(text)
   text = remove_stop_words(text)
   text = stemming(text)
   return text
 df['content'] = df['content'].apply(run)
 import re
 import string
```

```
  # remove all numbers with letters attached to them
  alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)
  # '[%s]' % re.escape(string.punctuation),' ' - replace punctuation with white space
# .lower() - convert all strings to lowercase
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())
# Remove all '\n' in the string and replace it with a space
remove_n = lambda x: re.sub("\n", " ", x)
# Remove all non-ascii characters
remove_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)
# Apply all the lambda functions wrote previously through .map on the comments column
df['content'] =
df['content'].map(alphanumeric).map(punc_lower).map(remove_n).map(remove_non_ascii)
def cleaner(text):
    text = text.lower()
    text = re.sub("@[^\s]+","",text)
    text = text.replace(":)","")
    text = text.replace("@","")
    text = text.replace("#","")
    text = text.replace(":(","")
    text = text.replace("|","")
return text
def remove_stop_words(text):
    sw = stopwords.words("english")
    clean_words = []
    text = text.split()
    for word in text:
        if word not in sw:
            clean_words.append(word)
    return " ".join(clean_words)
def stemming(text):
    ps = PorterStemmer()
    text = text.split()
    stemmed_words = []
```

```
for word in text :
            stemmed_words.append(ps.stem(word))
      return " ".join(stemmed_words)
def run(text):
    text = cleaner(text)
    text = remove_stop_words(text)
    text = stemming(text)
    return text
df['content'] = df['content'].apply(run)
data = df

comment_words = ''
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in data.content:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()



# Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
```

```
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)
# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
sample = df['content'][1]
print(sample)

testimonial = TextBlob(sample)
pola = testimonial.sentiment.polarity
subj = testimonial.sentiment.subjectivity
print('pola', pola, 'subj', subj)
def polarity(text):
    testimonial = TextBlob(text)
    polarity = testimonial.sentiment.polarity
    return polarity

def subjectivity(text):
    testimonial = TextBlob(text)
    subjectivity = testimonial.subjectivity
    return subjectivity
def senti(text, polarity_threshold=0.2):
    testimonial = TextBlob(text)
    senti = testimonial.sentiment.polarity

    if senti >= polarity_threshold:
        return 'Positive'
    elif np.abs(senti) < polarity_threshold:
```

```
            return 'Neutral'
        else:
            return 'Negative'

df['polarity'] = df['content'].apply(lambda x: polarity(x))
df['subjectivity'] = df['content'].apply(lambda x: subjectivity(x))
df['sentiment'] = df['content'].apply(lambda x: senti(x))
df.head()

fig, ax = plt.subplots(3, 1, figsize=(15, 15))
sentiments = df['sentiment'].unique()
for i, senti in enumerate(sentiments):
    senti_df = df.query('sentiment==@senti')
    cloud = ' '.join([tweet for tweet in senti_df['content']])
    wc = WordCloud(max_words=2000, width=1600, \
                   height=800, stopwords=stopwords).generate(cloud)
    ax[i].set_title(senti, fontsize=25)
    ax[i].axis('off')
    ax[i].imshow(wc , interpolation = 'bilinear')
plt.tight_layout()
plt.show()
```

Model Building

```
from sklearn import preprocessing
from sklearn.feature_selection import SelectFromModel
from sklearn.model_selection import train_test_split, KFold, cross_val_score
    from sklearn.metrics import f1_score, precision_score, recall_score,
    precision_recall_curve,    fbeta_score, confusion_matrix
from sklearn.metrics import roc_auc_score, roc_curve
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

```
from sklearn.metrics import accuracy_score

df['score'].unique

y = df['score']

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

from sklearn import metrics

def evaluate(model, X_train, X_test, y_train, y_test):

    y_test_pred = model.predict(X_test)

    y_train_pred = model.predict(X_train)

    print("TRAINIG RESULTS: \n==============================")

      clf_report = pd.DataFrame(classification_report(y_train, y_train_pred,
    output_dict=True))

    print(f"CONFUSION MATRIX:\n{confusion_matrix(y_train, y_train_pred)}")

    print(f"ACCURACY SCORE:\n{accuracy_score(y_train, y_train_pred):.4f}")

    print(f"CLASSIFICATION REPORT:\n{clf_report}")

    print("TESTING RESULTS: \n==============================")

    clf_report = pd.DataFrame(classification_report(y_test, y_test_pred, output_dict=True))

    print(f"CONFUSION MATRIX:\n{confusion_matrix(y_test, y_test_pred)}")

    print(f"ACCURACY SCORE:\n{accuracy_score(y_test, y_test_pred):.4f}")

print(f"CLASSIFICATION REPORT:\n{clf_report}")
```

Accuracy Comparison

```
scores = [lr_score,svm_score,neigh_score,gnb_score,clf_score,dt_score]

Model = ('Logistic Regression','SVM','KNeighbors Classifier','Naïve Bayes','Random
Forest Classifier','Decision Tree Classifier') y_pos = np.arange(len(Model))

print(y_pos)

print(scores)

import matplotlib.pyplot as plt2

plt2.barh(y_pos, scores, align='center', alpha=0.5,color='blue')

plt2.yticks(y_pos, Model)

plt2.xlabel('Score')

plt2.title('Accuracy comparison')

plt2.show()
```

4. SCREEN SHOTS

5.1 Training Result of Logistic Regression

```
TRAINIG RESULTS:
==============================
CONFUSION MATRIX:
[[ 707  130  863   29   30]
 [ 255  184 1207   63   69]
 [ 239  152 2617  221  232]
 [  51   57 1092  221  315]
 [  35   17  773  226  706]]
ACCURACY SCORE:
0.4227
CLASSIFICATION REPORT:
                    1            2           3           4            5
precision    0.549340     0.340741     0.39942    0.290789     0.522189
recall       0.401933     0.103487     0.75614    0.127304     0.401821
f1-score     0.464215     0.158758     0.52272    0.177083     0.454165
support    1759.000000  1778.000000  3461.00000  1736.000000  1757.000000

              accuracy     macro avg  weighted avg
precision     0.422743      0.420496      0.417197
recall        0.422743      0.358137      0.422743
f1-score      0.422743      0.355388      0.382551
support       0.422743  10491.000000  10491.000000
TESTING RESULTS:
==============================
CONFUSION MATRIX:
...
precision     0.671369      0.755198      0.724031
recall        0.671369      0.628966      0.671369
f1-score      0.671369      0.655501      0.660916
support       0.671369   2623.000000   2623.000000
```

Screenshot 5.1 Training Result of Logistic Regression

5.2 Training Result of SVM(Support Vector Machine)

```
TRAINIG RESULTS:
==============================
CONFUSION MATRIX:
[[ 545   29 1119   22   44]
 [ 154  136 1383   37   68]
 [ 115   26 2920  159  241]
 [  30    5 1204  203  294]
 [  19    5  908  182  643]]
ACCURACY SCORE:
0.4239
CLASSIFICATION REPORT:
                    1           2           3           4           5
precision    0.631518    0.676617    0.387576    0.336650    0.498450
recall       0.309835    0.076490    0.843687    0.116935    0.365965
f1-score     0.415713    0.137443    0.531151    0.173578    0.422054
support    1759.000000 1778.000000 3461.000000 1736.000000 1757.000000

              accuracy     macro avg  weighted avg
precision     0.423887      0.506162      0.487605
recall        0.423887      0.342583      0.423887
f1-score      0.423887      0.335988      0.367630
support       0.423887  10491.000000  10491.000000
TESTING RESULTS:
==============================
CONFUSION MATRIX:
...
precision     0.864659      0.891808      0.875125
recall        0.864659      0.848857      0.864659
f1-score      0.864659      0.864657      0.864142
support       0.864659   2623.000000   2623.000000
```

Screenshot 5.2 Training Result of Support Vector Machine

5.3 Training Result Naive Bias

```
TRAINIG RESULTS:
================================
CONFUSION MATRIX:
[[ 803  194    0  659  103]
 [   0  866    0  818   94]
 [  79  282 1078 1794  228]
 [   0    0    0 1733    3]
 [   0    1    0  887  869]]
ACCURACY SCORE:
0.5099
CLASSIFICATION REPORT:
                    1            2            3            4            5
precision    0.910431     0.644825     1.000000     0.294178     0.670008
recall       0.456509     0.487064     0.311471     0.998272     0.494593
f1-score     0.608103     0.554950     0.474994     0.454438     0.569090
support    1759.000000  1778.000000  3461.000000  1736.000000  1757.000000

              accuracy     macro avg  weighted avg
precision     0.509866      0.703888      0.752725
recall        0.509866      0.549582      0.509866
f1-score      0.509866      0.532315      0.523220
support       0.509866  10491.000000  10491.000000
TESTING RESULTS:
================================
CONFUSION MATRIX:
...
precision     0.278689      0.352231      0.377481
recall        0.278689      0.296799      0.278689
f1-score      0.278689      0.262560      0.259392
support       0.278689   2623.000000   2623.000000
```

Screenshot 5.3 Training Result of Naive Bias

5.4 Training Result of KNN Classifier

```
TRAINIG RESULTS:
==============================
CONFUSION MATRIX:
[[ 916  236   47   22  538]
 [ 800  281   91   38  568]
 [1388  460  363  120 1130]
 [ 653  233  176   86  588]
 [ 661  240  182   88  586]]
ACCURACY SCORE:
0.2128
CLASSIFICATION REPORT:
                    1            2            3            4            5
precision    0.207334     0.193793     0.422584     0.242938     0.171848
recall       0.520750     0.158043     0.104883     0.049539     0.333523
f1-score     0.296584     0.174102     0.168056     0.082297     0.226824
support    1759.000000  1778.000000  3461.000000  1736.000000  1757.000000

           accuracy     macro avg  weighted avg
precision  0.212754      0.247699      0.275999
recall     0.212754      0.233348      0.212754
f1-score   0.212754      0.189572      0.186282
support    0.212754  10491.000000  10491.000000
TESTING RESULTS:
==============================
CONFUSION MATRIX:
...
precision  0.508959     0.557824      0.586154
recall     0.508959     0.518130      0.508959
f1-score   0.508959     0.487655      0.499812
support    0.508959  2623.000000   2623.000000
```

Screenshot 5.4 Training Result of KNN Classifier

5.5 Training Result of Random Forest

```
TRAINIG RESULTS:
================================
CONFUSION MATRIX:
[[   0    0 1759    0    0]
 [   0    0 1778    0    0]
 [   0    0 3461    0    0]
 [   0    0 1736    0    0]
 [   0    0 1757    0    0]]
ACCURACY SCORE:
0.3299
CLASSIFICATION REPORT:
                1       2           3       4       5  accuracy
precision     0.0     0.0    0.329902     0.0     0.0  0.329902
recall        0.0     0.0    1.000000     0.0     0.0  0.329902
f1-score      0.0     0.0    0.496130     0.0     0.0  0.329902
support    1759.0  1778.0 3461.000000  1736.0  1757.0  0.329902

              macro avg  weighted avg
precision       0.065980      0.108835
recall          0.200000      0.329902
f1-score        0.099226      0.163674
support     10491.000000  10491.000000
TESTING RESULTS:
================================
CONFUSION MATRIX:
...
precision      0.105755
recall         0.325200
f1-score       0.159606
support     2623.000000
```

Screenshot 5.5 Training Result of Random Forest

5.6 Training Result of Decision Tree

```
TRAINIG RESULTS:
==============================
CONFUSION MATRIX:
[[1705    0   43    6    5]
 [  15 1668   73   10   12]
 [  22   12 3341   30   56]
 [   3    9  174 1487   63]
 [   6    6  180   56 1509]]
ACCURACY SCORE:
0.9256
CLASSIFICATION REPORT:
                    1            2            3            4            5
precision    0.973729     0.984071     0.876673     0.935809     0.917325
recall       0.969301     0.938133     0.965328     0.856567     0.858850
f1-score     0.971510     0.960553     0.918867     0.894436     0.887125
support    1759.000000  1778.000000  3461.000000  1736.000000  1757.000000

              accuracy     macro avg  weighted avg
precision     0.925555      0.937521      0.927742
recall        0.925555      0.917636      0.925555
f1-score      0.925555      0.926498      0.925400
support       0.925555  10491.000000  10491.000000
TESTING RESULTS:
==============================
CONFUSION MATRIX:
...
precision  0.475029     0.462555      0.468313
recall     0.475029     0.443685      0.475029
f1-score   0.475029     0.448555      0.466233
support    0.475029  2623.000000   2623.000000
```

Screenshot 5.6 Training Result of Decision Tree Classifier

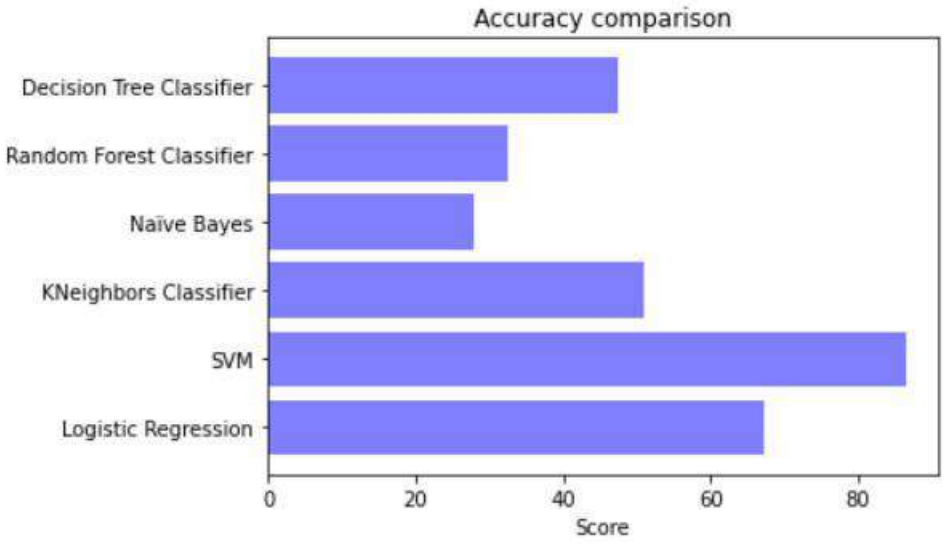
5.7 Educational Apps Ranking Result

Output:

Rank	Name
1	Byjus
2	KhanAcademy
3	Unacademy
4	WhiteHatJr
5	Udemy
6	Vedantu
7	Extramarks
8	Gradeup
9	Toppr
10	TestBook

Screenshot 5.7 Educational Apps Ranking Result

5.8 Accuracy Comparison Result



Screenshot 5.7 Accuracy Comparison Result

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover very conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that specified by the business and technical requirements,functions tested are available as system doc umentation, and user manuals.

Functional testing is centered on the following items: functions tested are availa ble as system documentation, and user

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requireme nts, key-functions, or special test cases. In addition, systematic coverage pertaini ng to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

S.NO	INPUT	If available	If not available
1	Upload dataset	Dataset loaded	There is no process
2	Data exploration	Visualize data to uncover insights	We cant explore
3	Data preprocessing	Process of transforming raw data into understandable data	We cant process
4	Model generation	Algorithm’s accuracy displayed	Algorithm not executed
5	Accuracy comparison graph	Comparison graph displayed	Prediction not done

7. CONCLUSION & FUTURESCOPE

7.1 PROJECT CONCLUSION

This study helps in Automating the process of evaluating educational apps which helps the students in choosing a good application for the purpose of online education. The iPAC framework is a well-established pedagogical framework for evaluating education apps along the dimensions: personalization, authenticity, and collaboration. We extended the initial keyword base of the iPAC framework with a data-driven approach based on online user reviews. Based on these keywords, we introduced a machine learning approach to identify and rank iPAC-based apps automatically. We achieved promising classification results, including an F1 score of 72%.

7.2 FUTURE SCOPE

Furthermore, we were able to show a moderate positive Spearman's rank correlation of 0.54 between the domain experts' app ranking and our feature-based app ranking. Our qualitative insights into identified iPAC-based apps and app reviews showed that our approach could capture iPAC-based app features as well as user feedback on the iPAC dimensions. We suggest a user-interface example of an education app search tool and showcase potential user scenarios for teachers, students, and vendors. We explain how our approach could enable the development of this tool. Thereby, this article fosters the mutual understanding between app vendors and teachers about textual app data and user feedback in app stores and beyond.

8. BIBILOGRAPHY

8.1 REFERENCES

[1] C.-K. Looi, D. Sun, and W. Xie, “Exploring students’ progression in an inquiry science curriculum enabled by mobile learning,” IEEE Trans. Learn. Technol., vol. 8, no. 1, pp. 43–54, Jan. 2015.

[2] M. G. Domingo and A. B. Gargante, “Exploring the use of educational technology in primary education: Teachers’ perception of mobile technology learning impacts and applications’ use in the classroom,” Comput. Human Behav., vol. 56, pp. 21–28, Mar. 2016.

[3] M. Kearney, K. Burden, and S. Schuck, “Disrupting education using smart mobile pedagogies,” in Didactics of Smart Pedagogy, L. Daniela, Ed. Cham, Switzerland: Springer, 2019, pp. 139–157.

[4] P. Sweeney and C. Moore, “Mobile apps for learning vocabulary: Categories, evaluation and design criteria for teachers and developers,” Int. J. Comput.-Assist. Lang. Learn. Teaching, vol. 2, no. 4, pp. 1–16, Oct. 2012.

[5] Y.-C. Hsu and Y.-H. Ching, “A review of models and frameworks for designing mobile learning experiences and environments,” Can. J. Learn. Technol., vol. 41, no. 3, pp. 1–22, 2015.

[6] M. Bano, D. Zowghi, M. Kearney, S. Schuck, and P. Aubusson, “Mobile learning for science and mathematics school education: A systematic review of empirical evidence,” Comput. Educ., vol. 121, pp. 30–58, Jun. 2018.

8.2 WEBSITES

[1] https://stackoverflow.com/questions/18754276/python-for-beginners

[2] https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization

[3] https://www.w3schools.com/python/python_ml_train_test.asp

International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211
Volume 10, Issue 6, June-2022, Impact Factor: 7.429, Available online at: www.ijaresm.com

Evaluation of Education Apps with App Store Data

B. P. Deepak Kumar[1], P. Rakesh[2], V. Anand Prakash[3], Mukul Chaubey[4]

[1]Assistant Professor, Department of CSE, CMR Technical Campus
[2,3,4]Department of CSE, CMR Technical Campus

----------**********----------

ABSTRACT

In recent days, due to rise in the number of applications that are available in application stores and the complexity of the features they provide, choosing a good app for learning has become challenging for teachers,students and parents. To assist people in selecting a good application there are several evaluation frameworks which are proposed in literature. One of the most well established framework is the iPAC framework which highlights the learner's experience in terms of personalization,authenticity and collaboration. But most of the literature methods are non-automatic and requires human effort hence we are proposing a system which uses natural language processing and machine learning techniques. These techniques uses data that is collected from the apps reviews and from the description provided by the application owner which are available on the application store. We evaluate the app based on the positive and negative reviews that are provided by the users and validating them by the positive and negative keywords. We used several classification algorithms to determine a single algorithm which can give us good accuracy results.

Key Words: App Description, App Reviews, Machine Learning, Natural language processing.

INTRODUCTION

Educational mobile apps offer innovative opportunities for teachers to improve students' learning. Over the past decade, researchers have investigated the effectiveness of apps in various education domains and proposed new pedagogical frameworks for mobile learning. However, with the vast, continuously increasing number of available apps, choosing "the right app" is becoming more and more difficult and time-consuming for teachers and student. One particular challenge is to efficiently select an app that appropriately supports the desired learning activities, assessment strategies, and pedagogical preferences.Teachers may use one of many existing digital frameworks to evaluate educational apps. However, most of them require a manual evaluation regarding different characteristics, which is time-consuming. Numerous frameworks have been proposed in the literature, ranging from complex multilevel models to smaller frameworks that often omit important sociocultural characteristics of mobile learning. Common themes include interactivity, control, communication, mobility of learners, and portability of mobile learning devices. The theoretical underpinning for this study is a robust and validated mobile pedagogical framework called iPAC. Grounded in sociocultural theory, it focuses on three distinctive mobile learning evaluation dimensions: personalization, authenticity, and collaboration.

LITERATURE SURVEY

Disrupting Education UsingSmart Mobile Pedagogies: Smart Pedagogy for Technology Enhanced Learning

As mobile technologies become more multifaceted and ubiquitous in society, educational researchers are investigating the use of these technologies in education. A growing body of evidence shows that traditional pedagogies still dominate the educational field and are misaligned with the diverse learning opportunities offered by the use of mobile technologies. There is an imperative to question those traditional notions of education, including how, where and when teaching and learning are enacted, and to explore the possible mediating roles of new mobile technologies. New smart pedagogies, which embrace the affordances offered by mobile technologies, have the potential to disrupt notions of schooling.

Mobile apps for learning vocabulary: Categories, evaluation and design criteria for teachers and developers

In this article the authors discuss the potential for mobile devices, mainly smartphones and tablets, to be used for language learning, offering frameworks for users to apply, including the categorisation of the different contexts and applications apps. They suggest critical success factors, including the importance of the user interface design and a taxonomy of interactivity and mobile "affordances" for publishers, developers, and users to consider when either evaluating existing apps, or developing their own mobile learning materials. The authors illustrate how these frameworks and taxonomies can work by applying them to the area of vocabulary learning. The article includes two case studies of app development projects in which one of the authors has had direct involvement to explore the relative

benefits and dis-benefits of re-versioning existing CDROM-based materials against developing an entirely new mobile learning app. Finally, they discuss the potential chasm between those interested in the potential of mobile language learning, including developers with insufficient knowledge of pedagogy, and language teachers who know about pedagogy, but have little interest in mobile learning. The article concludes with recommendations about how to overcome this divide with suggestions on how developers could make their language learning apps more pedagogically useful.

A review of models and frameworks fordesigning mobile learning experiences and environments

Mobile learning has become increasingly popular in the past decade due to the unprecedented technological affordances achieved through the advancement of mobile computing, making ubiquitous and situated learning possible. At the same time, there have been research and implementation projects whose efforts centered on developing mobile learning experiences for various learners' profiles, accompanied by the development of models and frameworks for designing mobile learning experiences. This paper focuses on categorizing and synthesizing models and frameworks targeted specifically on mobile learning. A total of 17 papers were reviewed, and the models or frameworks were divided into five categories and discussed: 1) pedagogies and learning environment design; 2) platform/system design; 3) technology acceptance; (4) evaluation; and 5) psychological construct. This paper provides a review and synthesis of the models/frameworks. The categorization and analysis can also help inform evaluation, design, and development of curriculum and environments for meaningful mobile learning experiences for learners of various demographics.

Mobile learning for science and mathematics school education: A systematic review of empirical evidence

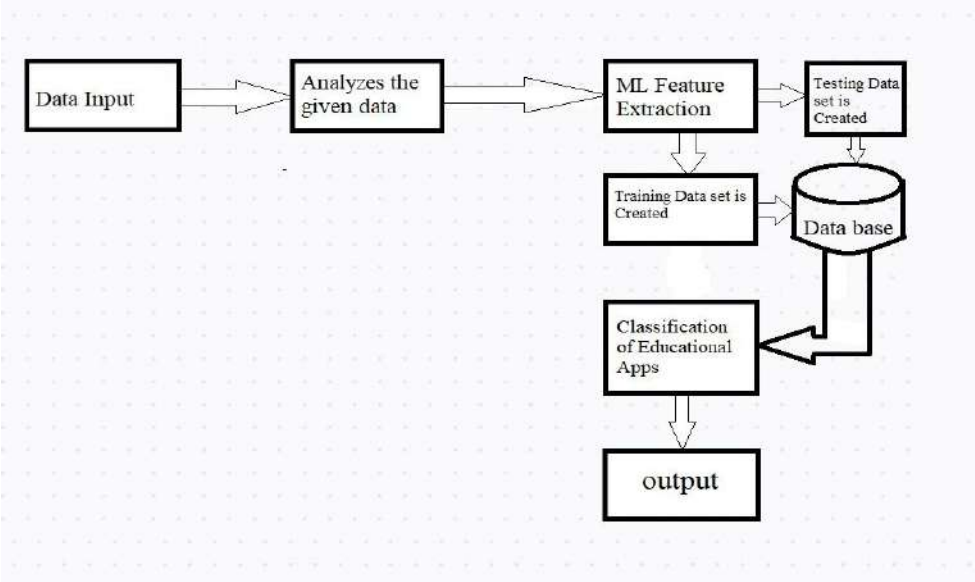
The ubiquity, flexibility, ease of access and diverse capabilities of mobile technologies make them valuable and a necessity in current times. However, they are under-utilized assets in mathematics and science school education. This article analyses the high quality empirical evidence on mobile learning in secondary school science and mathematics education. Our study employed a Systematic Literature Review (SLR) using well-accepted and robust guidelines. The SLR resulted in the detailed analysis of 49 studies (60 papers) published during 2003–2016. Content and thematic analyses were used to ascertain pedagogical approaches, methodological designs, foci, and intended and achieved outcomes of the studies. The apps and technologies used in these studies were further classified for domain, type and context of use. The review has highlighted gaps in existing literature on the topic and has provided insights that have implications for future research.

PROPOSED SYSTEM

In this article, we introduce an approach to automate the identification and comparison of iPAC relevant apps. We experiment with natural language processing and machine learning techniques, using data from the app description and app reviews publicly available in app stores. We further empirically validate the keyword base of the iPAC framework based on the app users' language in app reviews.

METHODOLOGY

A. System Architecture



B. Modules

Data Input: The data that is necessary for the purpose of our study is given.

Analyzing Data : The given data is preprocessed and is sent to the machine learning algorithm.

ML Feature Extraction: Ml feature extraction is performed create training data and test data after creating the training data the algorithm is trained on that data.

Database: Database is used to store the data and to get the when required.

Classification: Classification of the applications based on the reviews will be done and the application with most positive reviews is found.

Output: Results are given.

C. Methods

1. **Logistic Regression :** Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.Linear regression is used for generating continuous values like the price of the house, income, population, etc. In logistic regression, we generally compute the probability which lies between the interval 0 and 1 (inclusive of both). Then probability can be used to classify the data.

2. **SVM :** Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

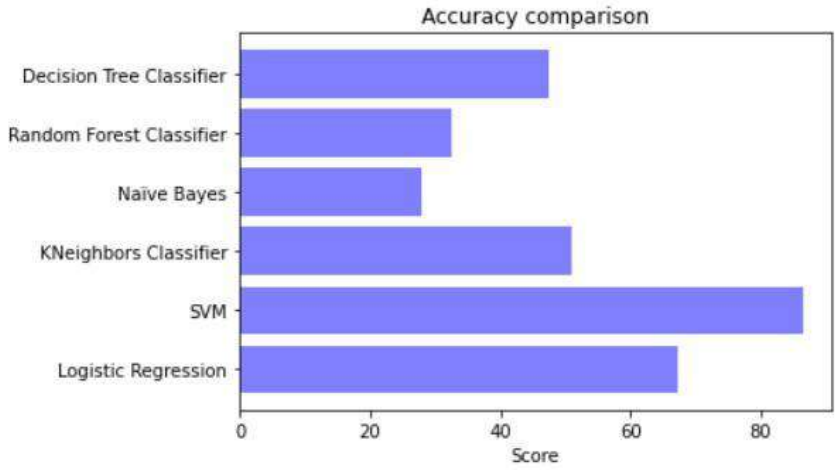
3. **Knn classifier :** This article concerns one of the supervised ML classification algorithm-KNN(K Nearest Neighbors) algorithm. It is one of the simplest and widely used classification algorithms in which a new data point is classified based on similarity in the specific group of neighboring data points. This gives a competitive result.KNN aims for pattern recognition tasks. K-Nearest Neighbor also known as KNN is a supervised learning algorithm that can be used for regression as well as classification problems. Generally, it is used for classification problems in machine learning.

4. **Naïve Bayes :** Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets. It can be used for Binary as well as Multi-class Classifications. It performs well in Multi-class predictions as compared to the other Algorithms. It is the most popular choice for text classification problems.Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

5. **Random Forest Classifier:** Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.It can perform both regression and classification tasks. A random forest produces good predictions that can be understood easily. It can handle large datasets efficiently. The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

OUTPUT

Accuracy comparison results



Output fig 1: Accuracy comparison bar graph result

CONCLUSION AND FUTURE WORK

This study helps in Automating the process of evaluating educational apps which helps the students in choosing a good application for the purpose of online education. The iPAC framework is a well-established pedagogical framework for evaluating education apps along the dimensions: personalization, authenticity, and collaboration. We suggest a user-interface example of an education app search tool and showcase potential user scenarios for teachers, students, and vendors. We explain how our approach could enable the development of this tool. Thereby, this article fosters the mutual understanding between app vendors and teachers about textual app data and user feedback in app stores and beyond.

ACKNOWLEDGMENT

We are grateful to our guide B.P. Deepak Kumar for helping us in completing this paper successfully. We also thank the department of CSE of CMR Technical Campus for their support.

REFERENCES

[1] C.-K. Looi, D. Sun, and W. Xie, “Exploring students’ progression in an inquiry science curriculum enabled by mobile learning,” IEEE Trans. Learn. Technol., vol. 8, no. 1, pp. 43–54, Jan. 2015.
[2] M. G. Domingo and A. B. Gargante, “Exploring the use of educational technology in primary education: Teachers’ perception of mobile technology learning impacts and applications’ use in the classroom,” Comput. Human Behav., vol. 56, pp. 21–28, Mar. 2016.
[3] M. Kearney, K. Burden, and S. Schuck, “Disrupting education using smart mobile pedagogies,” in Didactics of Smart Pedagogy, L. Daniela, Ed. Cham, Switzerland: Springer, 2019, pp. 139–157.
[4] P. Sweeney and C. Moore, “Mobile apps for learning vocabulary: Categories, evaluation and design criteria for teachers and developers,” Int. J. Comput.-Assist. Lang. Learn. Teaching, vol. 2, no. 4, pp. 1–16, Oct. 2012.
[5] Y.-C. Hsu and Y.-H. Ching, “A review of models and frameworks for designing mobile learning experiences and environments,” Can. J. Learn. Technol., vol. 41, no. 3, pp. 1–22, 2015.
[6] M. Bano, D. Zowghi, M. Kearney, S. Schuck, and P. Aubusson, “Mobile learning for science and mathematics school education: A systematic review of empirical evidence,” Comput. Educ., vol. 121, pp. 30–58, Jun. 2018.
[7] J. Clement, “Number of apps available in leading app stores as of 1st quarter 2020,” 2020. [Online]. Available: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/
[8] M. Bano, D. Zowghi, and M. Kearney, “Feature based sentiment analysis for evaluating the mobile pedagogical affordances of apps,” in Tomorrow’s Learning: Involving Everyone. Learning With and About Technologies and Computing, A. Tatnall and M. Webb, Eds. vol. 515, Cham, Switzerland: Springer, 2017, pp. 281–291.
[9] M. Stevenson, J. Hedberg, K. Highfield, and M. Diao, “Visualizing solutions: Apps as cognitive stepping-stones in the learning process,” Electron. J. E-Learn., vol. 13, no. 5, pp. 366–379, 2015.
[10] R. Kay and J. Kwak, “Creating an evidence-based framework for selecting and evaluating mathematics apps,” in Proc. Soc. Inf. Technol. Teacher Educ. Int. Conf., 2018, pp. 755–760.

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

UGC Journal No. : 7647

Certificate of Publication

P. Rakesh

Department of CSE, CMR Technical Campus

TITLE OF PAPER

Evaluation of Education Apps with App Store Data

has been published in

IJARESM, Impact Factor: 7.429, Volume 10 Issue 6, June- 2022

Paper Id: IJ-1706221689

Certificate No.: 186221404

Date: 18-06-2022

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

UGC Journal No. : 7647

Certificate of Publication

V. Anand Prakash

Department of CSE, CMR Technical Campus

TITLE OF PAPER

Evaluation of Education Apps with App Store Data

has been published in

IJARESM, Impact Factor: 7.429, Volume 10 Issue 6, June- 2022

Paper Id: IJ-1706221689

Certificate No.: 186221406

Date: 18-06-2022

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed Multi-disciplinary Journal

UGC Journal No. : 7647

Certificate of Publication

Mukul Chaubey

Department of CSE, CMR Technical Campus

TITLE OF PAPER

Evaluation of Education Apps with App Store Data

has been published in

IJARESM, Impact Factor: 7.429, Volume 10 Issue 6, June- 2022

Paper Id: IJ-1706221689

Certificate No.: 186221410

Date: 18-06-2022

Website: www.ijaresm.com

Email: editor.ijaresm@gmail.com

Authorized Signatory