

# CLASSIFICATION OF SONGS



Artificial intelligence & Machine learning  
MACHINE LEARNING PROJECT

Presented by:-  
C205 DIVANSHU UPPAL  
C209 KESHAVI PANDEY  
C214 RISHIRAJ MISHRA

In the guidance of:-  
Prof. Varsha Nemade

AIM



- Classification of songs- we have made this project
- considering the search results of the listeners.
  - This project will basically help the listeners to sort their personalised playlist according to their mood.

ACCESSING SPOTIFY CREDENTIALS/SCRAPING DATA

LIBRARIES/READING IN DATASETS/GRAPHING

REFINING DATASETS

CREATING THE MODEL

STANDARDIZING DATA

SPLITTING DATA INTO TRAINING AND TESTING SETS

TRAINING THE MODEL

MAKING PREDICTIONS AND EVALUATING THE PREDICTIONS

ADJUSTING THE K-VALUE FOR A STRONGER MODEL

APPLYING SVM,DECISION TREE & RANDOM FOREST















User

Choice of emotions











>

beautiful

Title
 <b>You Don't Even Know Me</b> Faouzia
 <b>This Mountain</b> Faouzia
 <b>Tears of Gold</b> Faouzia
 <b>Out Of Love</b> Alessia Cara
 <b>I'm Not The Only One</b> Sam Smith
 <b>Can't Help Falling in Love</b> Haley Reinhart
 <b>Before You Go</b> Lewis Capaldi
 <b>Hold Me While You Wait</b> Lewis Capaldi
 <b>Grace</b> Lewis Capaldi
 <b>Maybe</b> Lewis Capaldi

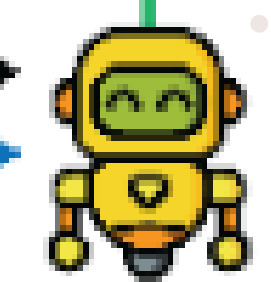
>

Happy

Title
 <b>Sharara</b> Jeet-Pritam, Asha Bhosle
 <b>Billo</b> J Star
 <b>Bad Boy (From "Saa</b> Badshah, Neeti Mohan
 <b>Brown Rang</b> Yo Yo Honey Singh
 <b>Paris Ka Trip</b> Yo Yo Honey Singh, Mithoon
 <b>Love Dose</b> Yo Yo Honey Singh
 <b>Shringaar (feat. Mithoon)</b> Vayu, Raftaar, Aastha
 <b>Besharam Rang</b> Vishal-Shekhar, Shilpa
 <b>Ik Tera</b> Maninder Buttar
 <b>Sandal</b> Sunanda Sharma



Request music based on specific features



Agent


Recommendation

A MODEL THAT CAN TAKE ANY SONG AND, DEPENDING ON ITS MUSICAL ATTRIBUTES, CLASSIFY IT INTO A PLAYLIST WITH THE NEAREST, SHARED QUALITIES.

# DATASET

	artist	track	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	type	id
0	Mamta Sharma	Munni Badnaam	0.696	0.95	1	-4.942	1	0.0963	0.0873	0.000345	0.0944	0.743	125.029	audio_features	6fORBVEcJNDJ1gdq5uu0Azspotify:track:6fORBVEcJNDJ1gdq5uu0Az
1	Jeet-Pritam	Sharara	0.602	0.891	6	-8.655	0	0.449	0.384	0.00234	0.335	0.548	94.975	audio_features	1akzSqvnTzAxlyWYqTZcJXspotify:track:1akzSqvnTzAxlyWYqTZcJX
2	J Star	Billo	0.895	0.812	1	-4.025	1	0.0379	0.15	0	0.357	0.961	129.994	audio_features	7nt4c7z8RyiFHZmORGqwkkspotify:track:7nt4c7z8RyiFHZmORGqwkk
3	Badshah	Bad Boy (From "Saaho")	0.892	0.848	8	-4.818	1	0.183	0.242	0.013	0.0992	0.523	105.013	audio_features	5l3Xih0C2ID3fqngBrBbGvspotify:track:5l3Xih0C2ID3fqngBrBbGv
4	Yo Yo Honey Singh	Brown Rang	0.804	0.902	11	-3.067	0	0.0846	0.265	0.000413	0.0651	0.72	155.98	audio_features	3JblyrSNumxS5LHpAvLYF5spotify:track:3JblyrSNumxS5LHpAvLYF5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
195	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
196	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
197	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
198	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
199	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
200 rows × 16 columns															


[illegible]







# beautiful


sameeha

PREVIEW






1	Holy - Acoustic	3:25
2	You Don't Even Know Me	2:59
3	This Mountain	2:22
4	Tears of Gold	2:56
5	Out Of Love	3:47
6	I'm Not The Only One	3:59







# Happy

keshavi

PREVIEW





1	Munni Badnaam	5:05
2	Sharara	4:56
3	Billo	3:15
4	Bad Boy (From "Saaho")	2:58
5	Brown Rang	3:31
6	Paris Ka Trip	3:16

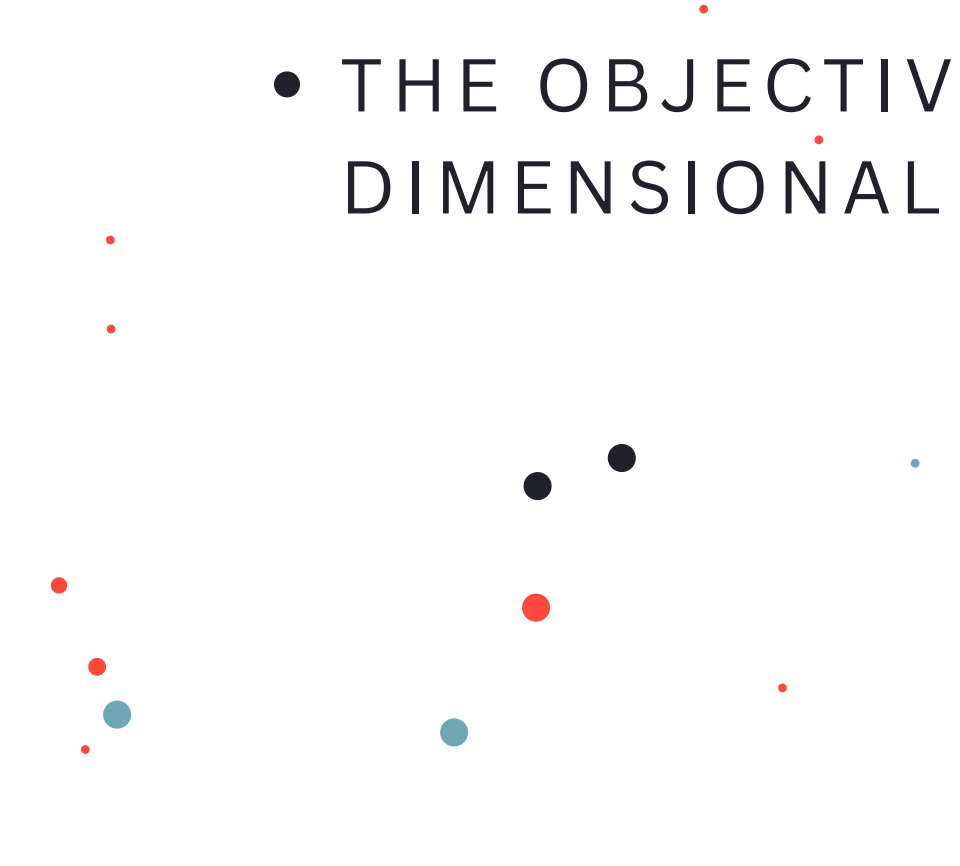
# • K NEAREST NEIGHBOUR (KNN)

- KNN IS A SIMPLE ALGORITHM USED IN MACHINE LEARNING FOR CLASSIFICATION AND REGRESSION.
- IT WORKS BY FINDING THE K DATA POINTS (NEAREST NEIGHBORS) IN THE TRAINING SET THAT ARE CLOSEST IN DISTANCE TO THE NEW DATA POINT.
- KNN DOES NOT REQUIRE EXPLICIT TRAINING OF THE MODEL, BUT REQUIRES CHOOSING K, DISTANCE METRIC, AND HYPERPARAMETERS FOR OPTIMIZATION.
- KNN IS WIDELY USED IN IMAGE CLASSIFICATION, RECOMMENDATION SYSTEMS, AND ANOMALY DETECTION..



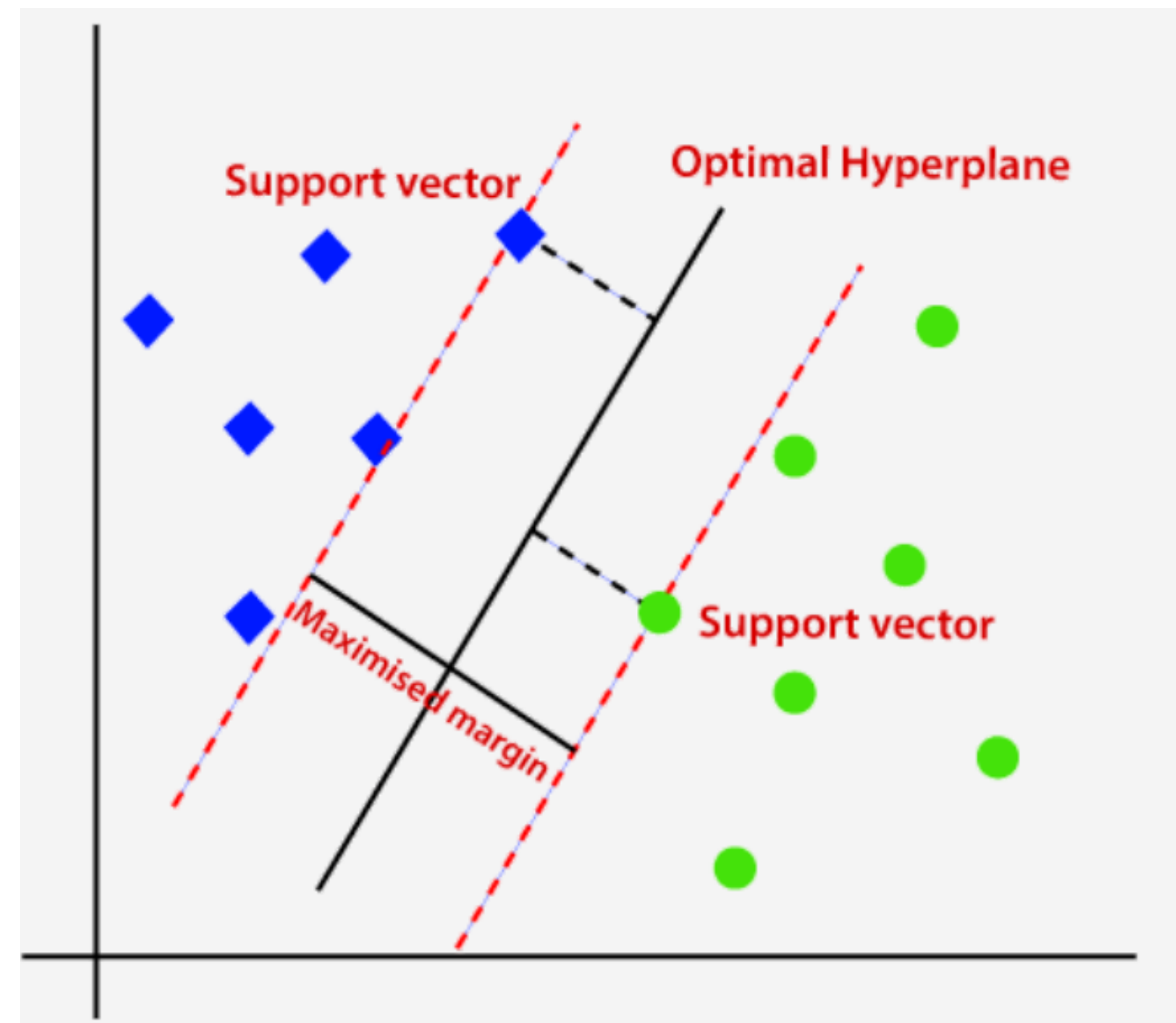


# SUPPORT VECTOR MACHINES

- 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.
- 



- THE DIMENSION OF THE HYPERPLANE DEPENDS UPON THE NUMBER OF FEATURES.
- IF THE NUMBER OF INPUT FEATURES IS TWO, THEN THE HYPERPLANE IS JUST A LINE.
- IF THE NUMBER OF INPUT FEATURES IS THREE, THEN THE HYPERPLANE BECOMES A 2-D PLANE. IT BECOMES DIFFICULT TO IMAGINE WHEN THE NUMBER OF FEATURES EXCEEDS THREE



# Random Forest

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output

- Decision Tree is the most powerful and popular tool for classification and prediction.
- A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.
- The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting

# Accuracy

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$



# ACCURACY

## K NEAREST NEIGHBOUR (KNN)

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    knn = KNeighborsClassifier(n_neighbors=3,p=2,metric='euclidean')
    knn.fit(X_train, y_train)
```

KNeighborsClassifier  
KNeighborsClassifier(metric='euclidean', n\_neighbors=3)

```
[ ] prediction = knn.predict(X_test)
    prediction
```

```
array([0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
        1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
        1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0,
        1, 0, 0, 0, 0, 1, 0, 0, 0, 0])
```

```
▶ from sklearn.metrics import accuracy_score
   print(accuracy_score(y_test, prediction))
```

0.925

# RANDOM FOREST

```
[ ] from sklearn.ensemble import RandomForestClassifier  
    clf = RandomForestClassifier(n_estimators=100)  
    clf.fit(X_train,y_train)  
    y_pred_1 = clf.predict(X_test)
```

```
[ ] print("accuracy_score",accuracy_score(y_test,y_pred_1))
```

```
accuracy_score 0.9666666666666667
```

# SUPPORT VECTOR MACHINE (SVM)

```
[ ] from sklearn import svm  
    clf= svm.SVC(kernel='poly')  
    clf.fit(X_train,y_train)  
    y_pred = clf.predict(X_test)
```

```
[ ] print("accuracy score",accuracy_score(y_test,y_pred))
```

```
accuracy score 0.7
```

```
[ ] from sklearn import svm  
    clf= svm.SVC(kernel='rbf')  
    clf.fit(X_train,y_train)  
    y_pred = clf.predict(X_test)  
    print("accuracy score",accuracy_score(y_test,y_pred))
```

```
accuracy score 0.6916666666666667
```



# FUTURE SCOPE

- Improve the recommendations by incorporating user listening history and real-time user feedback
- It helps the listeners to sort the songs more easily according to the mood with the help of the algorithm

# CONCLUSION

- Spotify uses a combination of KNN, SVM, and Random Forest algorithms to classify its songs into different categories. by analyzing the acoustic and the textual features of each song, Spotify can group similar songs together and recommend them to users based on their listening history and preferences.
- This classification system is essential for providing users with personalized playlists and improving their overall listening experience on the platform.
- As Spotify continues to grow and add more songs to its collection, these algorithms will become even more critical for maintaining a high level of user engagement.



THANK YOU