

Pengenalan pola mengetahui kategori destinasi wisata favorit di India dengan metode *Support Vector Machine* (SVM)

Mohamad Rifky Fajri

Program Studi Informatika, Universitas Ahmad Dahlan, Yogyakarta Indonesia

INFORMASI ARTIKEL

ABSTRAK

Histori Artikel

Diterima : 25 September 2021

Direvisi : 25 Januari 2022

Diterbitkan : 15 April 2022

Kata Kunci:

Pengenalan Pola

Wisata

Support Vector Machine

Sistem yang menentukan kategori destinasi wisata mana yang paling banyak dikunjungi oleh para wisatawan luar negeri dan dalam negeri. Sistem ini menggunakan metode classification untuk menentukan mana yang paling banyak diminati oleh pengunjung, menggunakan dataset dari hasil riset ke beberapa wisatawan yang berkunjung ke india untuk mengetahui kategori destinasi yang dituju. Ini juga menjadi keuntungan untuk wisatawan baru yang akan berkunjung ke india untuk memilih kategori destinasi mana yang menjadi rekomendasi saat kita berkunjung ke india. Metode yang dipakai adalah classification SVM (*Support Vector Machine*) untuk mengetahui pola dari destinasi wisata disana. Dengan memakai metode ini kita dapat melakukan perjalanan pariwisata tanpa kecewa karena tidak salah memilih tempat wisata

2022 SAKTI – Sains, Aplikasi, Komputasi dan Teknologi Informasi.

Hak Cipta.

I. Pendahuluan

Disaat masa liburan seperti ini kita biasanya melakukan perjalan pariwisata entah di dalam negeri maupun di luar negeri. Tapi seringkali kita kebingungan memilih tujuan destinasi wisata apa yang akan kita pilih apalagi jika kita baru pertama melakukan perjalanan wisata. Kondisi pariwisata saat ini sedang tidak bagus karena adanya covid-19 pasti destinasi pariwisata tersebut kosong. Ini menjadi konsen kita setelah pandemic ini berakhir pastilah banyak orang yang akan pergi ke tempat-tempat wisata. Kami menampilkan hasil berdasarkan kategori objek wisata dengan harapan bisa membantu untuk memilih objek wisata apa yang akan dilakukan setelah covid-19. Penelitian ini menggugulkan *Support Vector Machine* (SVM) untuk mengetahui destinasi mana yang paling menarik di suatu negara lebih spesifiknya di india karena *dataset* yang kita pakai adalah kategori destinasi di india. Jika kita sudah mengetahui destinasi mana yang paling banyak dikunjungi itu menjadi rekomendasi kita. Sehingga kita tidak kebingungan lagi menentukan apa destinasi wisata kita di india.

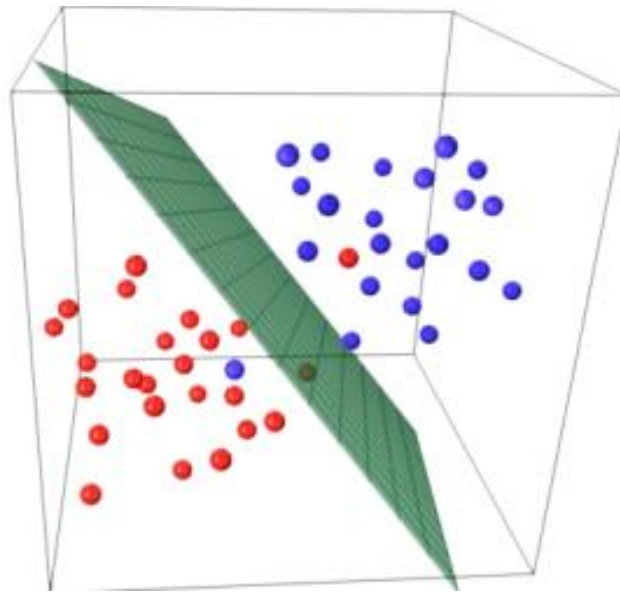
II. Material dan Metode

A. *Classification*

Classification adalah metode yang paling umum pada data mining. Persoalan bisnis seperti Churn Analysis, dan *Risk Management* biasanya melibatkan metode *Classification*. *Classification* adalah tindakan untuk memberikan kelompok pada setiap keadaan. Setiap keadaan berisi sekelompok atribut, salah satunya adalah *class attribute*. Metode ini butuh untuk menemukan sebuah model yang dapat menjelaskan class attribute itu sebagai fungsi dari *input attribute*[1,2].

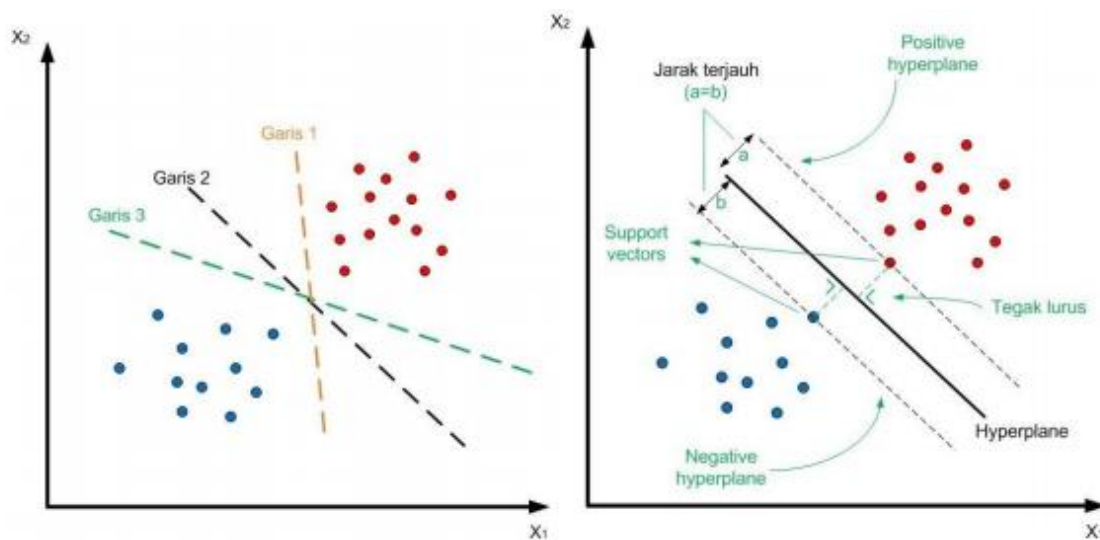
B. *Support Vector Machine* (SVM)

SVM merupakan mesin pembelajaran dengan model pembelajaran terbimbing. Algoritma SVM menganalisa data untuk regresi dan klasifikasi. Ide awal dari SVM adalah menemukan *hiperple* terbaik yang membagi data menjadi 2 kelas [3], seperti gambar 1. di bawah ini.



Gambar 1. Algoritma SVM

Hiperplen adalah 2 titik yang membentuk garis pembatas antara 2 kelas. Pembagian 2 kelas ini harus sebaik mungkin dimana garis batasnya bisa memisahkan 2 kelompok dengan jarak terjauh antara titik terluar di masing-masing kelompok dengan garis pembatas itu sendiri [4].



Gambar 2. Pembagian menjad 2 Kelas Data

Pada gambar 2 sebelah kiri ada 3 kemungkinan garis yang membagi 2 kelas data, namun garis terbaik untuk membagi data tersebut adalah garis 2. Pada dasarnya SVM hanya dapat membagi data menjadi 2 kelompok, apabila ingin membuat klasifikasi dalam 3 kelompok maka harus menggunakan multi SVM [5.6]. Memakai metode diatas untuk membandingkan dua kategori destinasi wisata yang menurut survei menjadi destinasi favorit disana. Jika kita lihat di dataset ada 6 kategori bandingkan dua-dua karena tidak bisa dibandingkan sekaligus karena harus memakai multi SVM [7].

C. Akuisisi Data

Dataset buddymove berjumlah 249 baris dan 7 kolom yaitu kolom sports, religious, nature, theatre, shopping, dan picnic, saya ambil di <https://archive.ics.uci.edu/ml/datasets/BuddyMove+Data+Set>, x_data training = 186, 2 x_data test = 63, 2 y_data training = 186 y_data test 63.

III. Hasil dan Pembahasan

Sebelumnya kita ambil terlebih dahulu dataset di UCI atau Kaggle lalu kita proses dataset tersebut, Berikut langkah-langkahnya :

```
In [1]: from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
```

Gambar 3. Import Library

Pertama kita import dulu library yang akan digunakan seperti gambar 3.

```
In [3]: # Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]]
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num=None, figsize=(6 * nGraphPerRow, 8 * nGraphRow), dpi=80, facecolor='w', edgecolor='k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
            plt.ylabel('counts')
            plt.xticks(rotation=90)
            plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad=1.0, w_pad=1.0, h_pad=1.0)
    plt.show()
```

```
In [4]: # Correlation matrix
def plotCorrelationMatrix(df, graphWidth):
    filename = df.dataframeName
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]]
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NaN or constant columns ({df.shape[1]} is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w', edgecolor='k')
    corrMat = plt.matshow(corr, fignum=1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()
```

```
In [5]: # Scatter and density plots
def plotScatterMatrix(df, plotSize, textSize):
    df = df.select_dtypes(include=[np.number])
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]]
    columnNames = list(df)
    if len(columnNames) > 10:
        columnNames = columnNames[:10]
    df = df[columnNames]
    ax = pd.plotting.scatter_matrix(df, alpha=0.75, figsize=[plotSize, plotSize], diagonal='kde')
    corrs = df.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k=1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes fraction', ha='center', va='center', size=8)
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

Gambar 4. Algoritma Diagram dan Plot

Lakukan langkah-langkah diatas untuk membuat diagram serta plot untuk menampilkan perbandingan antara kategori destinasi wisata yang nanti akan kita bandingkan.

```
In [6]: nRowsRead = 1000 # specify 'None' if want to read whole file
df1 = pd.read_csv('buddymove_holidayiq.csv', delimiter=',', nrows = nRowsRead)
df1.dataframeName = 'buddymove_holidayiq.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')

There are 249 rows and 7 columns
```

Gambar 5. Pengolahan Dataset

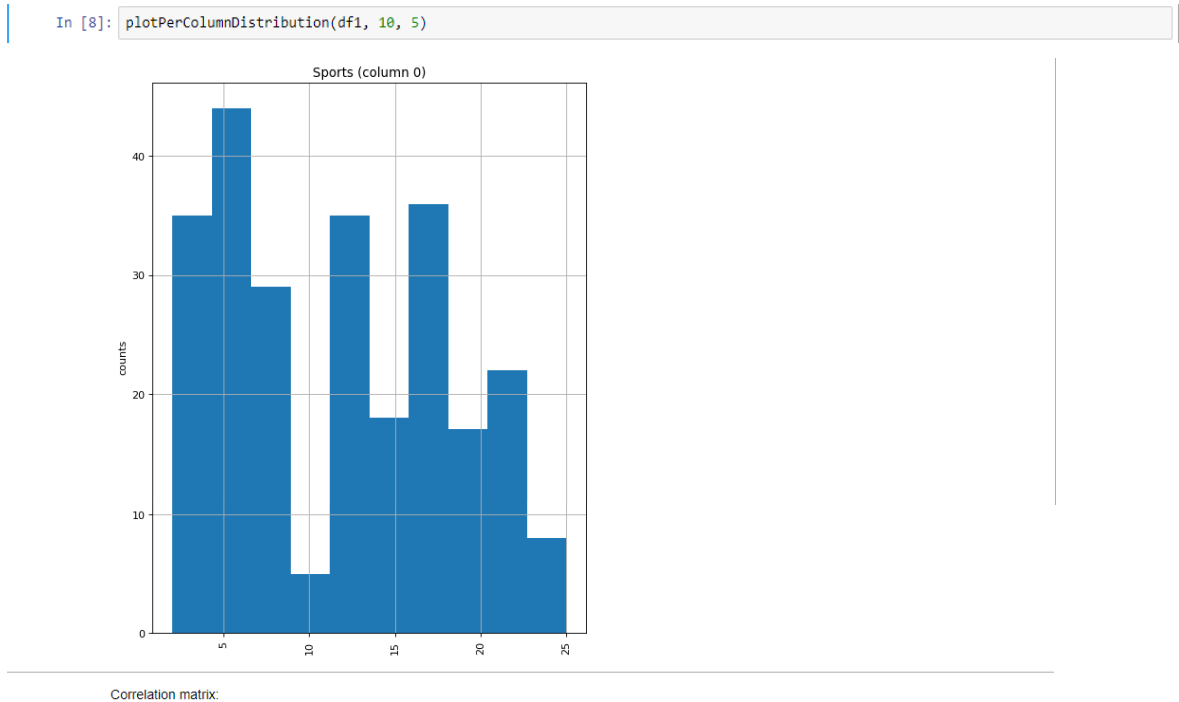
Lalu setelah itu kita masukan dataset yang sudah kita ambil dari UCI atau Kaggle tadi dengan cara seperti gambar 5 didalam dataset tersebut ada 249 baris dan 7 kolom.

```
In [7]: df1.head(5)
Out[7]:
```

	UserID	Sports	Religious	Nature	Theatre	Shopping	Picnic
0	User 1	2	77	79	69	68	95
1	User 2	2	62	76	76	69	68
2	User 3	2	50	97	87	50	75
3	User 4	2	68	77	95	76	61
4	User 5	2	98	54	59	95	86

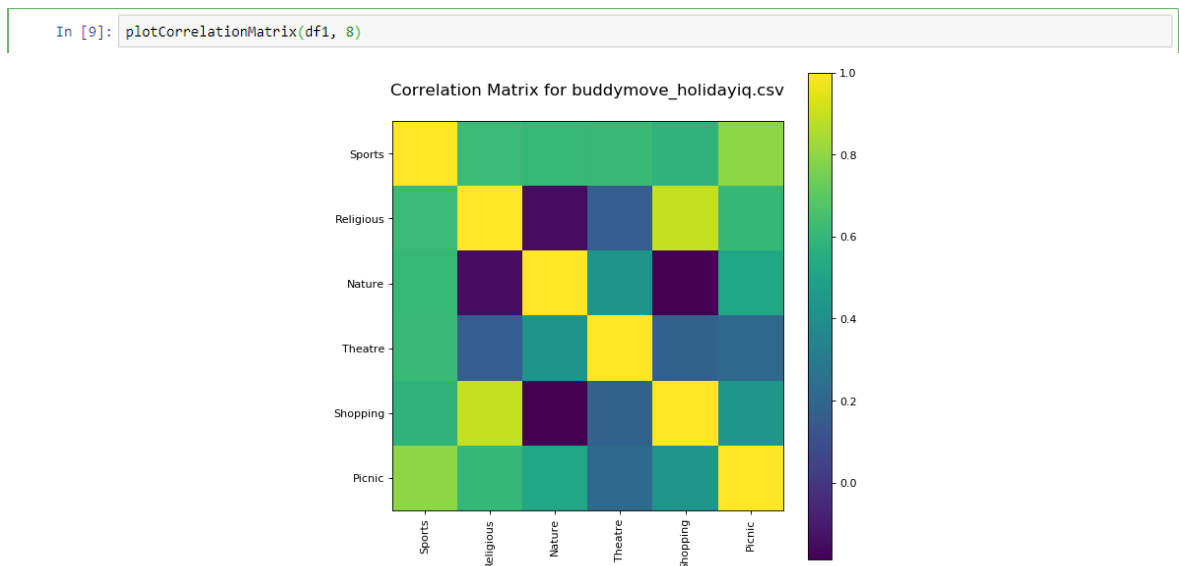
Gambar 6. Data Tabel 7 Kolom

Berikut adalah sebagian isi didalam datasetnya.



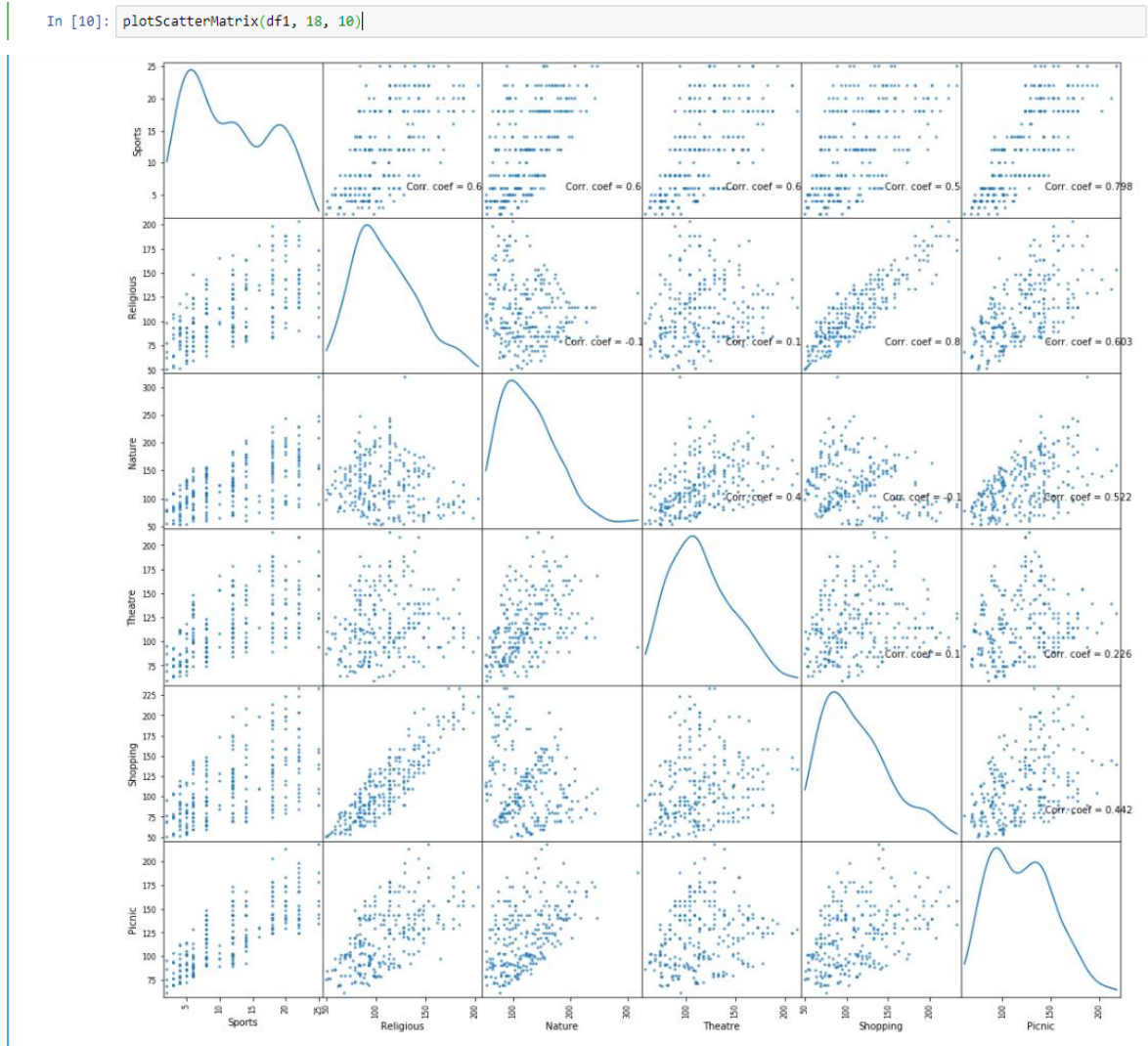
Gambar 7. Data berdasarkan Kolom Sport

Gambar 7. diatas menunjukan plot berdasarkan kolom, kita ambil contoh kolom sports



Gambar 8. Plot Correlation Matrik Kolom Dataset

Gambar 8. diatas menunjukkan plot correlation matrix seluruh kolom yang ada di dataset



Gambar 9. Hasil Perbandingan Plot

Gambar 9. diatas adalah hasil dari perbandingan semuanya. Jika dilihat dari plotnya menghasilkan plot yang berbeda-beda bisa disimpulkan bahwa selera wisatawan pun berbeda dan jika kita lihat lagi yang lumayan stabil plotnya ialah sports

```
In [12]: from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)

In [13]: classifier.fit(x_train, y_train)

Out[13]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
kernel='linear', max_iter=-1, probability=False, random_state=0,
shrinking=True, tol=0.001, verbose=False)

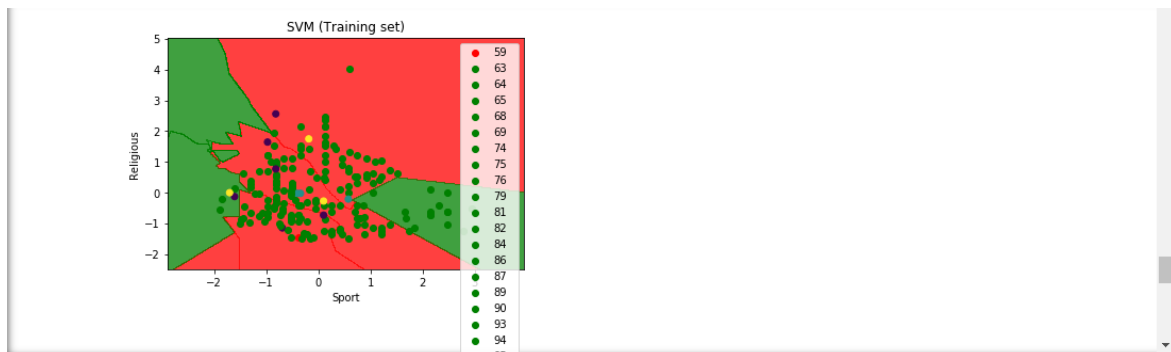
In [14]: y_pred = classifier.predict(x_test)

In [15]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

Gambar 10. Tahapan SVM

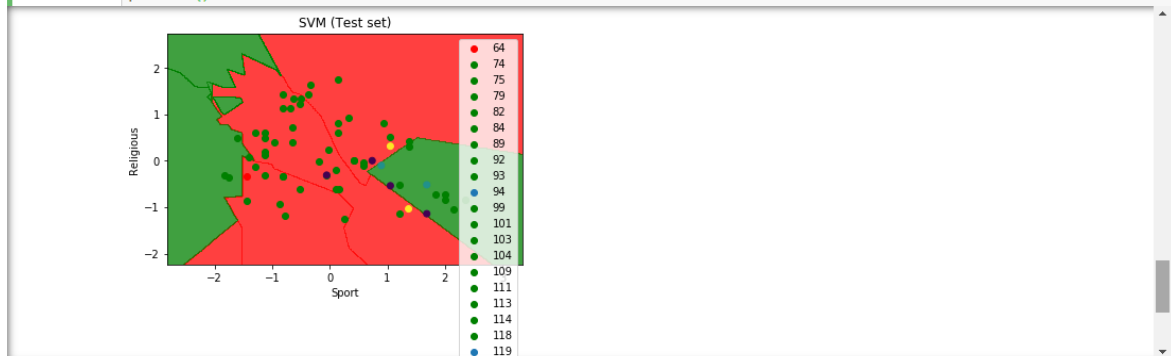
Gambar 10. diatas adalah tahapan SVM pertama import terlebih dahulu setelah itu kita classifikasikan. Disini kita memprediksi dengan mengambil data training dan data set yang sudah kita classifikasikan

```
In [16]: from matplotlib.colors import ListedColormap
x_set, y_set = x_train, y_train
X1, X2 = np.meshgrid(np.arange(start = x_set[:, 0].min() - 1, stop = x_set[:, 0].max() + 1, step = 0.01),
                    np.arange(start = x_set[:, 1].min() - 1, stop = x_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('SVM (Training set)')
plt.xlabel('Sport')
plt.ylabel('Religious')
plt.legend()
plt.show()
```



Gambar 12. Hasil Running Program SVM

```
In [36]: from matplotlib.colors import ListedColormap
x_set, y_set = x_test, y_test
X1, X2 = np.meshgrid(np.arange(start = x_set[:, 0].min() - 1, stop = x_set[:, 0].max() + 1, step = 0.01),
                    np.arange(start = x_set[:, 1].min() - 1, stop = x_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('SVM (Test set)')
plt.xlabel('Sport')
plt.ylabel('Religious')
plt.legend()
plt.show()
```



Gambar 13. Hasil dari data test

IV. Kesimpulan

Metode ini kita bisa menentukan tujuan destinasi apa yang mau kita kunjungi dengan mengetahui plot kategori destinasi wisata diatas tetapi kembali kepada selera orang yang berbeda-beda jika sukanya diolahraga ya memilih berkunjung di destinasi sports begitu juga dengan yang lain tetapi ini bisa jadi acuan untuk mengetahui selera wisatawan mancanegara terhadap destinasi disuatu negara terutama di india

Daftar Pustaka

- [1] Satria & Kartika, 2021, Classification of Color Features In Butterflies Using The Support Vector Machine (SVM)
- [2] Meiriyama, M. 2018. Klasifikasi Citra Buah Berbasis Fitur Warna Hue Saturation Value (HSV) Dengan klasifikator SVM. Jurnal Komputer Terapan, 4(1), 50-61.
- [3] Z. M. S. Muhammad Ichwan, Irma Amelia Dewi, 2019, “Klasifikasi Support Vector Machine (SVM) Untuk Menentukan Tingkat Kemanisan Mangga Berdasarkan Fitur Warna,” MIND J., Vol. 3, No. 2, pp. 16–23,
- [4] Christianini, N., & John S, T. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press.
- [5] Sembiring, K. (2007). Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan. Skripsi: Program Studi Teknik Informatika Institut Teknologi Bandung
- [6] Yohannes, Y., Pribadi, M. R., & Chandra, L. 2020. Klasifikasi Jenis Buah dan Sayuran Menggunakan SVM Dengan Fitur Saliency-HOG dan Color Moments. Elkha, 12(2), 125. <https://doi.org/10.26418/elkha.v12i2.42160>
- [7] B. Caraka, B. A. A. Sumbodo, and I. Candradewi, 2017, “Klasifikasi Sel Darah Putih Menggunakan Metode Support Vector Machine (SVM) Berbasis Pengolahan Citra Digital,” IJEIS (Indonesian J. Electron. Instrum. Syst., Vol. 7, No. 1, pp. 25–36, doi: 10.22146/ijeis.15420.

Lampiran

<https://archive.ics.uci.edu/ml/datasets/BuddyMove+Data+Set>

<https://www.kaggle.com/kerneler/starter-buddy-movedataset-24c5f175-3>