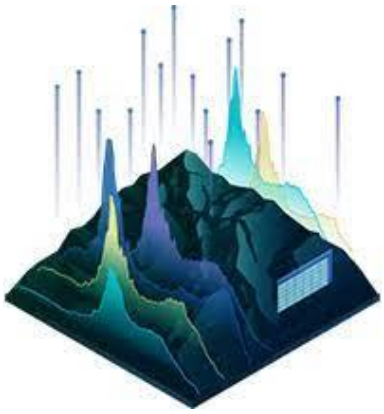
شرکت پژوهشی آراد

آزمون عملی

نام و نام خانوادگی	پیام صائمی
تاریخ	تابستان ۱۴۰۲
لینک‌پروژه	https://colab.research.google.com/drive/1apMkCxrdVGq-FJvedHLetudfw5aAXlBo



فهرست

۱.مسئله

تعدادی خودرو توسط موسسه‌های کرایه خودرو به افراد مختلف کرایه داده می‌شوند. تعداد و نوع خودروهایی که در هر موسسه وجود دارد مشخص است. به دلیل خرابی یا از کار افتادن خودروها لیست موجودی هر موسسه بروزرسانی میشود. همچنین ویژگی‌های هر خودرو در لیست دیگری وجود دارد که این ویژگی‌ها ثابت هستند. می‌خواهیم یک ماژول هوشمند طراحی کنیم که با استفاده ترکیب(از دو لیست موجود) ، بتواند با دریافت ویژگی‌های یک خودرو و همچنین موسسه کرایه‌دهنده آن، نوع خودروی مشاهده شده را تشخیص دهد .

۱–۱.فرض مسئله

دو فایل اکسل وجود دارد ، موجودی موسسه‌ها و مشخصات ماشین‌ها . جدول مشخصات ماشین‌ها همواره ثابت است ، اما جدول موجودی موسسه‌ها بروزرسانی میشود . بروزرسانی شدن جدول موسسه به این معنی است که با هربار تغییر این جدول ، باید آن را از ورودی بگیریم و ماژول هوشمند مجددا باید با دیتاهای جدید آموزش دیده شود (کافی است دیتای بروز شده را از ورودی بخوانید).

۲.مقدمه

طبق مسئله دو مجموعه داده‌ی جدا از هم وجود دارد که باید با الحاق این مجموعه‌ها، ماژول هوشمندی طراحی شود. از آنجایی که مسئله یک مسئله‌ی Classification به روش Supervised است ، باید از الگوریتم‌های مربوط به دسته‌بندی استفاده کرد. در این مسئله از الگوریتم KNN استفاده شده است.

۱–۲.الگوریتم KNN

الگوریتم K-Nearest Neighbors (KNN) یک الگوریتم یادگیری ماشین در حوزه تشخیص الگو است. این الگوریتم بر اساس اصول ساده عمل می‌کند. تعیین نزدیک‌ترین همسایه‌ها برای پیش‌بینی برچسب یا ویژگی جدید، KNNنزدیک‌ترین نقاط داده از داده‌های آموزش را مشخص می‌کند. با استفاده از برچسب‌های نزدیک‌ترین نقاط، برچسب جدید توسط رأی اکثریت مشخص می‌شود. به عبارت دیگر، اگر بیشتر نقاط نزدیک از یک دسته خاص باشند، نقطه جدید هم به همان دسته تعلق می‌گیرد. برچسب یا ویژگی جدید بر اساس برچسب رأی اکثریت انتخاب می‌شود. الگوریتم KNN بر پایه فاصله میان نقاط کار می‌کند. یعنی نقاطی که در فضای ویژگی‌ها نزدیک‌تر به یکدیگر هستند، احتمالاً در یک دسته مشابه قرار می‌گیرند. برای محاسبه فاصله، معیارهای مختلفی مانند منهتن دیستنس، فاصله یوکلیدی و... می‌توانند به کار رود .دقت و کارایی KNN به انتخاب درست پارامتر K و معیار فاصله، همچنین حذف داده‌های نویزی بستگی دارد.

۲–۲.مجموعه‌داده ها

۱–۲–۲.موجودی موسسه ها

مجموعه‌داده‌ای که ستون‌های آن شامل نام موسسه‌ها و سطرهای آن شامل نوع ماشین‌ها است. تقاطع سطرها و ستون‌ها نشان دهنده‌ی موجودی هر موسسه است. مجموعه داده شامل ۴ موسسه‌ی کرایه دهنده‌ی ماشین است و در مجموع ۹ نوع ماشین مختلف در این مجموعه‌داده وجود دارد .

۲–۲–۲.مشخصات ماشین‌ها

مجموعه‌داده‌ای که ستون‌های آن شامل ویژگی های ماشین‌ها است که سه ستون قیمت‌کرایه، سرعت و کلاس در این مجموعه‌داده وجود دارد و سطرهای آن شامل ۲۸۸ عدد ماشین است که در ۹ دسته تقسیم‌بندی میشود.

۳.فراخوانی دیتا بر روی دیتافریم

با استفاده از کتابخانه pandas مجموعه‌داده‌ها از google drive فراخوانی میشود.

```
instituteInventory_dataFrame = pd.read_csv('/content/drive/MyDrive/arad_csv/Instituteinventory.csv')
```

شکل . ۱ فراخوانی داده های موسسه

```
carFeatures_dataFrame = pd.read_csv('/content/drive/MyDrive/arad_csv/Carfeatures.csv')
```

شکل . ۲فراخوانی داده های ماشین ها

۴.قابل استفاده کردن دیتا

۴–۱. تنظیم تیتر و تبدیل پارامترها از نوع string به کلاس integer

برای آنکه مقدور باشد از داده‌های موجود برای آموزش ماژول استفاده شود باید از نوع string به integer تبدیل شود. که این کار با تابع factorize() صورت میگیرد. درواقع این تابع هر شئ را با اعداد شماره‌گذاری میکند. و همچنین تابع set_axis() تیترها را تنظیم می‌کند. در این عملیات شئ‌های موجود در ستون carname از صفر تا هشت شماره گذاری میشود .

```
instituteInventory_dataFrame = instituteInventory_dataFrame.set_axis(['Carname','first_Institute ','second_Institute ',
                                                                      'Third_Institute','Fourth_Institute'],
                                                                     axis=1)
instituteInventory_dataFrame['Carname'] = pd.factorize(instituteInventory_dataFrame['Carname'])[0]
```

شکل . ۳ شمارگذاری شئ‌های موجود در ستون Carnameدر مجموعه‌داده موسسه ها

```
carFeatures_dataFrame = carFeatures_dataFrame.set_axis(['Carname' , 'speed' , 'class' , 'Rental_price'] , axis=1)
carFeatures_dataFrame['Carname'] = pd.factorize(carFeatures_dataFrame['Carname'])[0]
carFeatures_dataFrame['class'] = pd.factorize(carFeatures_dataFrame['class'])[0]
```

شکل . ٤ شمارگذاری شئ‌های موجود در ستون‌های class و carname در مجموعه‌داده‌ی ماشین‌ها

۴–۲.نتیجه بخش ۴–۱

	Carname	first_Institute	second_Institute	Third_Institute	Fourth_Institute
0	0	0	0	0	4
1	1	2	0	0	0
2	2	1	2	3	0
3	3	0	2	0	0
4	4	0	0	1	2
5	5	0	2	2	0
6	6	1	0	4	2
7	7	1	0	0	4
8	8	0	2	0	2

شکل . ٥ نتیجه عملیات factorize و set_axis برروی مجموعه‌داده‌ي موسسه

	Carname	speed	class	Rental_price
0	0	200	0	1000
1	1	250	1	500
2	2	180	2	700
3	3	180	1	1000
4	4	150	2	500
...	...	...	...	...
283	4	110	2	530
284	5	250	1	900
285	6	200	3	800
286	7	300	0	950
287	8	200	0	2000

شکل . ٦ نتیجه عملیات factorize و set_axis برروی مجموعه‌داده‌ي ماشین‌ها

۵.الحاق دو مجموعه‌داده

۵–۱.ساختن آیدی برای هر ماشین

در مرحله‌ی قبل برای هر مدل ماشین (شئ) با استفاده از تابع factorize() عددی خاص به هر مدل ماشین اختصاص داده شد. حال به هر ماشین موجود در مجموعه‌داده ماشین‌ها شناسه‌ی یکتایی داده می‌شود که با استفاده از الگوریتم زیر عملی شده است. در واقع هدف از این عملیات آن است که بعد از الحاق دو مجموعه داده اطمینان حاصل شود که الحاق به درستی صورت گرفته باشد.

```
# Generate car dataFrame
carTypeID = 0
for car_id in car_ids:
    car_type = carTypes[carTypeID]
    car_data.append({'CarID': car_id, 'CarType': car_type,
                     'Feature1':carFeatures_values[car_id , 1 ],
                     'Feature2': carFeatures_values[car_id , 2 ],
                     'Feature3': carFeatures_values[car_id , 3 ]})

    carTypeID+=1
    if carTypeID == 9 :
      carTypeID=0

carDataFrame = pd.DataFrame(car_data)
```

شکل . ۷ تنظیم آیدی برای هر ماشین

۲-۵.نتیجه عملیات ۱-۵

	CarID	CarType	Feature1	Feature2	Feature3
0	0	0	200	0	1000
1	1	1	250	1	500
2	2	2	180	2	700
3	3	3	180	1	1000
4	4	4	150	2	500
...	...	...	...	...	...
283	283	4	110	2	530
284	284	5	250	1	900
285	285	6	200	3	800
286	286	7	300	0	950
287	287	8	200	0	2000

شکل . ۸ نتیجه عملیات ۱-۵

۳-۵.الحاق دو مجموعه با استفاده از ضرب کارتزین

با ایجاد کردن ستون DummyKey در دو مجموعه داده، رابطه‌ای بین آنها ایجاد می‌کنیم که با استفاده از آن بتوانیم ضرب کارتزین را اعمال کنیم و دو مجموعه‌داده را باهم الحاق کنیم .

ضرب کارتزین (Cartesian Product) مفهومی در ریاضیات است که در تئوری مجموعه‌ها مورد استفاده قرار می‌گیرد. دو مجموعه A و B را در نظر بگیرید، ضرب کارتژین آنها، مجموعه‌ای از تمام جفت‌های مرتب (a, b) است که a از مجموعه A و b از مجموعه B باشد. به عبارت دیگر، ضرب کارتژین دو مجموعه، تمام امکانات ترکیب عناصر از این دو مجموعه را نمایش می‌دهد .اگر $A = \{a_1, a_2, ...\}$ و $B = \{b_1, b_2, ...\}$ دو مجموعه باشند، ضرب کارتژین A و B به صورت زیر تعریف می‌شود $A \times B = \{(a_1, b_1), (a_1, b_2), ..., (a_2, b_1), (a_2, b_2), ...\}$: در اینجا، هر جفت (a, b) یک عنصر از ضرب کارتژین A و B است، که a از مجموعه Aو b از مجموعه B انتخاب شده است. اگر تعداد عناصر مجموعه A برابر با m و تعداد عناصر مجموعه B برابر با n باشد، تعداد اعضای ضرب کارتژین A و B برابر با m * n خواهد بود .

درواقع انجام این عملیات به این دلیل است که ابتدا همه‌ی ماشین‌ها در همه‌ی موسسه‌ها موجود شود و بعد از این عملیات سطرهایی که ماشین مورد نظر در موسسه متناظر وجود ندارد حذف میشوند .

```
# Create a dummy key to perform a Cartesian product
carDataFrame['DummyKey'] = 1
institutionsDataFrame['DummyKey'] = 1

# Merge the dataframes using the dummy key
mergedData = pd.merge(carDataFrame, institutionsDataFrame, on='DummyKey')

# Drop the dummy key column
mergedData.drop('DummyKey', axis=1, inplace=True)
```

شکل . ۹ اعمال ضرب کارتزین و الحاق دو مجموعه‌داده

۴–۵.نتیجه عملیات کارتزین

	CarID	CarType	Feature1	Feature2	Feature3	InstitutionID	InstitutionName
0	0	0	200	0	1000	1	first_Institute
1	0	0	200	0	1000	2	second_Institute
2	0	0	200	0	1000	3	Third_Institute
3	0	0	200	0	1000	4	Fourth_Institute
4	1	1	250	1	500	1	first_Institute
...	...	...	...	...	...	...	...
1147	286	7	300	0	950	4	Fourth_Institute
1148	287	8	200	0	2000	1	first_Institute
1149	287	8	200	0	2000	2	second_Institute
1150	287	8	200	0	2000	3	Third_Institute
1151	287	8	200	0	2000	4	Fourth_Institute

1152 rows × 7 columns

شکل . ۱۰ نتیجه الحاق با اعمال ضرب کارتزین

۶.پیش پردازش‌ها

۱–۶.آماده سازی دیتا

در این بخش ابتدا index هایی که مقادیر داخل آن صفر است (ماشین مورد نظر در موسسه متناظر موجود نیست) استخراج میشود . این Index ها در لیست zerosIndex ذخیره میشود .

```
# extracing zeros index
zerosIndex = []
rows , cols = instituteInventory_values.shape
for row in range (rows):
  for col in range(1 , cols):
    if instituteInventory_values[row,col] == 0 :
      zerosIndex.append([row,col])
```

شکل . ۱۱ استخراج Index مقادیر صفر در ماتریس موسسه‌ها

مبرهن است که در لیست zerosindex ستون اول نشان‌دهنده ماشین‌ها و ستون دوم نشان‌دهده موسسه‌ی متناظر با هر ماشینی است که موجودی آن ماشین در موسسه متناظرش صفر است . به عنوان مثال [۲ , ۰] یکی از مقادیر این ماتریس است که میگوید "تمام ماشین ها با شناسه صفر (همه‌ی بنزها) در موسسه ۲ وجود ندارد" پس تمام سطرهایی که در دیتافریم mergedData ، روبروی بنز ، موسسه‌ی دوم وجود دارد حذف میشود و همین روال بر تمام مقادیر اعمال میشود . بعد از این عملیات ستون‌های carID , instituteName نیز حذف میشود .

```
# Drop the lines that the existence of the car in the institution is not confirmed
for Index in zerosIndex :
  carname , instituteid = Index
  indexAge = mergedData[ (mergedData['CarType'] == carname) & (mergedData['InstitutionID'] == instituteid) ].index
  mergedData.drop(indexAge , inplace=True)
```

شکل . ۱۲ حذف سطرهایی که با موجودی موسسه تطابق ندارد

۲-۶. نتیجه بخش ۶-۱

	CarType	Feature1	Feature2	Feature3	InstitutionID
3	0	200	0	1000	4
4	1	250	1	500	1
8	2	180	2	700	1
9	2	180	2	700	2
10	2	180	2	700	3
...	...	...	...	...	...
1143	6	200	3	800	4
1144	7	300	0	950	1
1147	7	300	0	950	4
1149	8	200	0	2000	2
1151	8	200	0	2000	4

544 rows × 5 columns

شکل . ۱۳ دیتافریم ۵*۵۴۴ نشان دهنده دیتاهای مطابق با موجودی موسسه ها است

۲-۶.جدا کردن داده‌های آموزشی و تست

```
Xdata = mergedData_values[: , 1:5]
ydata = mergedData_values[: , 0]
```

شکل . ۱٤ جداسازی ویژگی‌ها و برچسب‌ها

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( Xdata, ydata, test_size=0.20  , random_state=10)
```

شکل . ۱۵ جداسازی داده‌های آموزشی و داده‌های تست

۷.آموزش مدل KNN

بررسی کردن مقادیر k از ۱ تا ۲۰ برای پیدا کردن بهترین k

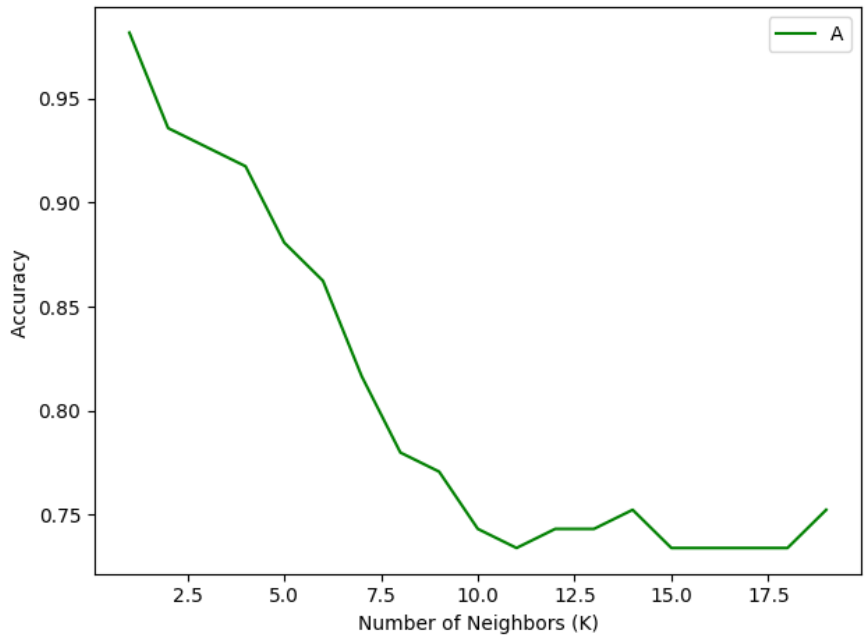
```
Ks = 20
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))
for n in range(1,Ks):
    neigh = KNeighborsClassifier(n_neighbors= n ).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test , yhat)
    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])
print (std_acc)
print(mean_acc)
```

شکل ۱۶

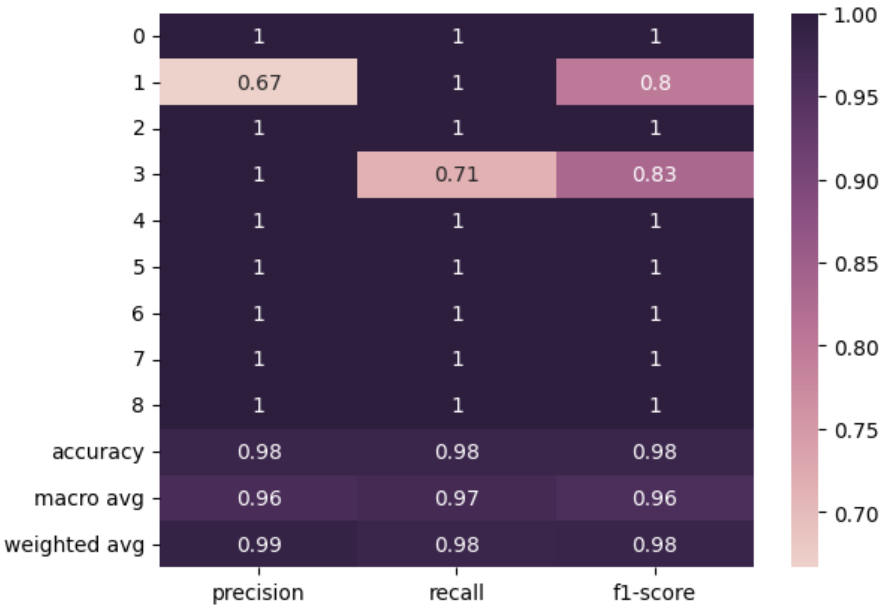
۸.نتایج نهایی

۸–۱.نمودار دقت براساس مقدار K

از آنجایی که نمودار کاملا نرولی است و در k=۱ بهترین دقت محاسبه شده است ، نشان‌دهنده است امر است که داده‌های هم‌کلاس شباهت زیادی به یکدیگر دارند .

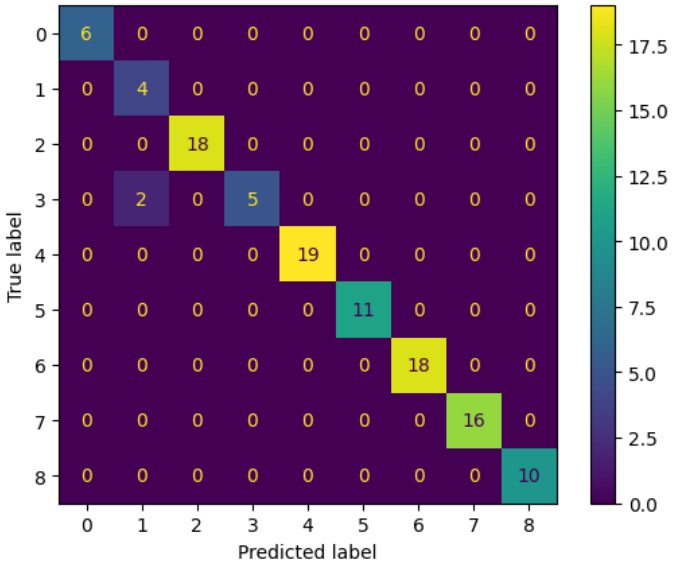


۲-۸. جدول classification_report



۳-۹.جدول confusion matrix

توزیع داده در ۹ کلاس بصورت تقریبی با اختلاف کمی برابر است و از هر کلاس تعدادی داده وجود دارد و در ۱۰۰ داده‌ی موجود در جدول فقط دوتای آنها اشتباه پیشبینی شده است که نتیجه‌ی نسبتا مطلوبی است.



پاینده باشید