بیتا آذری جو ۸۳۹۵۹۸۰۰۷

گزارش کار تمرین اول

۱,۱ تشریح الگوریتم KNN :

این الگوریتم یکی از الگوریتم های یادگیری با ناظر می باشد. به دلیل سادگی پیاده سازی به شدت مورد استفاده قرار می گیرد. این الگوریتم non-parametric بوده به این معنی که هیچ فرض اولیه ای راجع به داده ها نمی کند. این الگوریتم lazy learning هم می باشد به این معنی که هیچ فاز آموزش(train) ندارد و مستقیما با داده های تست کار می شود.
در این الگوریتم داده ها به صورت رندوم به دو بخش آموزش و تست تقسیم می شود. هر داده برداری از ویژگی های مربوط به آن نمونه دارد. برای هر داده ی تست فاصله ی اقلیدسی آن با سایر نقاط را به دست آورده و به صورت صعودی مرتب می کنیم. سپس در k نمونه ی اول می بینیم جواب نهایی متعلق به کدام کلاس بوده است. هر کلاسی که تعدادش بیشتر بود می گوییم داده ی تست متعلق به آن کلاس خواهد بود.

در تمرین برای مقایسه دقت با کتابخانه ی sklearn و انتخاب فاصله ی minkowski با پارامتر ۲ جواب زیر به دست آمد که نشان دهنده ی دقت خوب پیاده سازی می باشد.
پ.ن : در کد اصلی قسمت های مربوط به کتابخانه ی sklearn کامنت شده اند.

```
bita@bita-K401UQK:~/programming/KNN-Classifier$ python KNN.py
Best K for prediction is : 5
Manual Accuracy: 0.979

Sklearn Accuracy: 0.971
```

۱,۲ تشریح مفهوم میانگین و واریانس و چولگی :

میانگین : مقدار مرکزی تعدادی داده ی گسسته می باشد. در واقع جمع مقادیر داده تقسیم بر تعداد کل داده ها. اگر توزیع داده پیوسته باشد. به جای جمع انتگرال می گیریم و امید ریاضی را محاسبه می کنیم.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1 + \cdots + x_n)$$

حالت گسسته

$$\mathrm{E}[X] = \int_{-\infty}^{\infty} x f(x)\, \mathrm{d}\, x.$$

حالت پیوسته

به دلیل پراکندگی در مقادیر داده میانگین معیار خوبی برای مقایسه نمی باشد. معمولا از میانه استفاده می کنند.

واریانس : نوعی سنجش پراکندگی می باشد. مقدار واریانس با میانگین‌گیری از مربع فاصله مقدار محتمل یا مشاهده شده با مقدار موردانتظار محاسبه می‌شود. در مقایسه با میانگین می‌توان گفت که میانگین مکان توزیع را نشان می‌دهد، در حالی که واریانس مقیاسی است که نشان می‌دهد که داده‌ها حول میانگین چگونه پخش شده‌اند. واریانس کمتر بدین معنا است که انتظار می‌رود که اگر نمونه‌ای از توزیع مزبور انتخاب شود مقدار آن به میانگین نزدیک باشد .واحد واریانس مربع واحد کمیت اولیه می‌باشد.

$$\sigma^2 = \sum_{i=1}^{N} p(x_i)(x_i - \mu)^2$$

اگر ویژگی ای در داده دارای واریانس بالا باشد اتفاقا خوب است چون نشان دهنده ی تغییرات زیاد می باشد و این تغییرات می تواند در تصمیم گیری نهایی اثرگذار باشد.

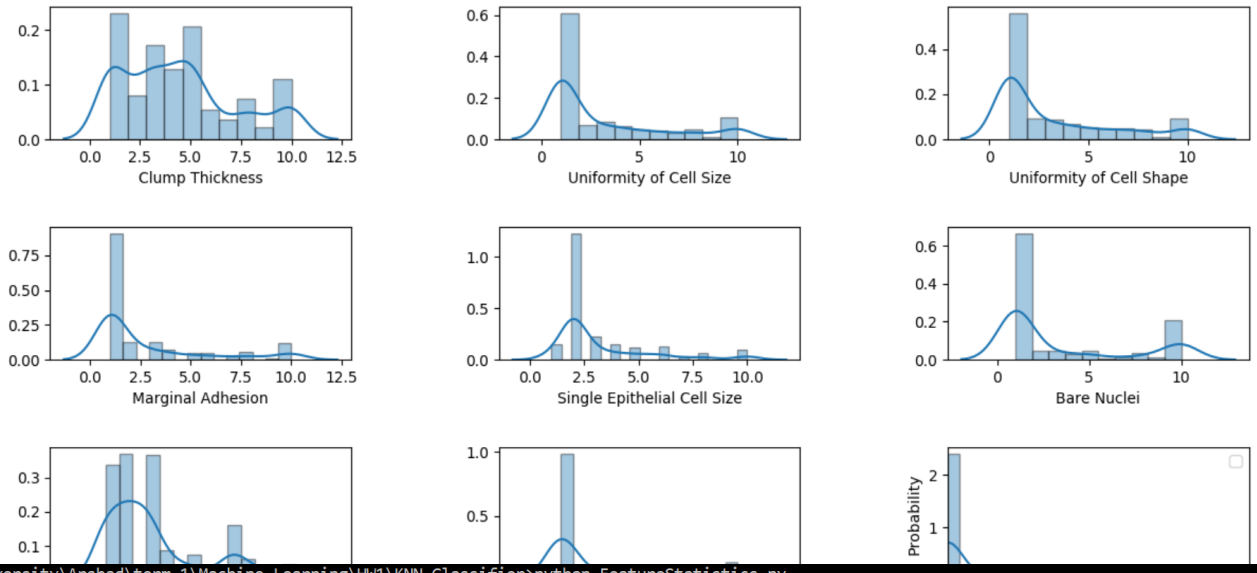
```
bita@bita-K401UQK:~/programming/KNN-Classifier$ python KNN.py
Best K for prediction is : 3
Manual Accuracy: 0.943
Manual Recall: 0.939
Manual Precison: 0.902
bita@bita-K401UQK:~/programming/KNN-Classifier$ python KNN.py
Number of features reduced to : 3
Best K for prediction is : 5
Manual Accuracy: 0.950
Manual Recall: 0.959
Manual Precison: 0.904
bita@bita-K401UQK:~/programming/KNN-Classifier$
```

شکل اول بدون PCA و شکل دوم دقت ها با استفاده از PCA به ازای همان داده ها می باشد

می بینیم که دقت ها حفظ شده اند.

چولگی : نشان دهنده ی میزان عدم تقارن توزیع احتمالی است. اگر داده ها نسبت به میانگین متقارن باشد، چولگی صفر است. در واقع فرمول چولگی گشتاور مرتبه ی سوم به انحراف معیار است.
در داده های موجود پیک داده ها به سمت چپ متمایل است بنابراین چولگی مثبت به دست می آید.

شکل توزیع احتمالی ویژگی ها به شرح زیر است :




```
D:\university\Arshad\term 1\Machine Learning\HW1\KNN-Classifier>python FeatureStatistics.py
Mean of Clump Thickness : 4.418
Variance of Clump Thickness : 7.928
Skewness of Clump Thickness : 0.593

Mean of Uniformity of Cell Size : 3.134
Vriance of Uniformity of Cell Size : 9.311
Skewness of Uniformity of Cell Size : 1.233

Mean of Uniformity of Cell Shape : 3.207
Vriance of Uniformity of Cell Shape : 8.832
Skewness of Uniformity of Cell Shape : 1.162
```

مقدار واریانس بزرگ در داده ها به این معنی است که پراکندگی بیشتری وجود دارد و هنگام استفاده از الگوریتم PCA داده هایی که واریانس بیشتری دارند شانس بیشتری هم برای انتخاب به عنوان principal component دارند.

۱,۳ تقسیم داده ها به صورت ۸۰-۲۰ برای داده و تست :

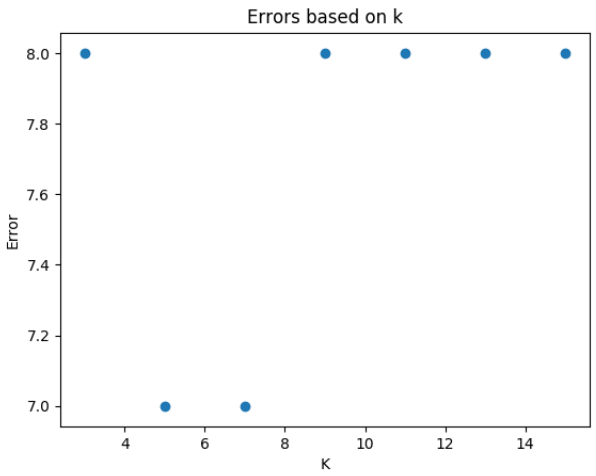
```
data=LoadData.loadDataset()

X = data.iloc[:,:-1].values    # Here first : means fetch all rows :-1 means except last column
y = data.iloc[:,10].values     # : is fetch all rows 10 means 10th column

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.20, random_state = 5)
```

۱,۴ انتخاب بهترین k و رسم خطای مدل :

به ازای random seed=5 و استفاده از PCA در کاهش بعد داده ی آموزش و تست، نتایج زیر به دست آمد:



۱,۵ محاسبه ی متریک های

ابتدا به تعریف های confusion matrix می پردازیم که برای محاسبه ی کارایی مدل کلاس بندی به کار می رود. سطر آن واقعیت و ستون مقدار پیش بینی است :

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

True Positive : اینکه فرد سرطان بدخیم داشته باشد و الگوریتم درست تشخیص دهد.

True Negative : اینکه فرد سرطان بدخیم داشته باشد و الگوریتم تشخیص ندهد.

False Positive : فرد سرطان بدخیم نداشته باشد ولی الگوریتم تشخیص دهد که سرطان بدخیم دارد.

False Negative : فرد سرطان بدخیم نداشته باشد و الگوریتم هم سرطان را شناسایی نکند.

دقت مقدار تشخیص درست سرطان (چه داشته باشد چه نداشته باشد) به تعداد کل نمونه های تست می باشد.

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

Recall میزان تشخیص درست و مثبت سرطان به کل تعداد نمونه هایی است که سرطان داشته اند.

$$\textit{Recall} = \frac{TP}{TP + FN}$$

Precision میزان تشخیص درست و مثبت سرطان به کل نمونه هایی است که الگوریتم مثبت تشخیص داده.

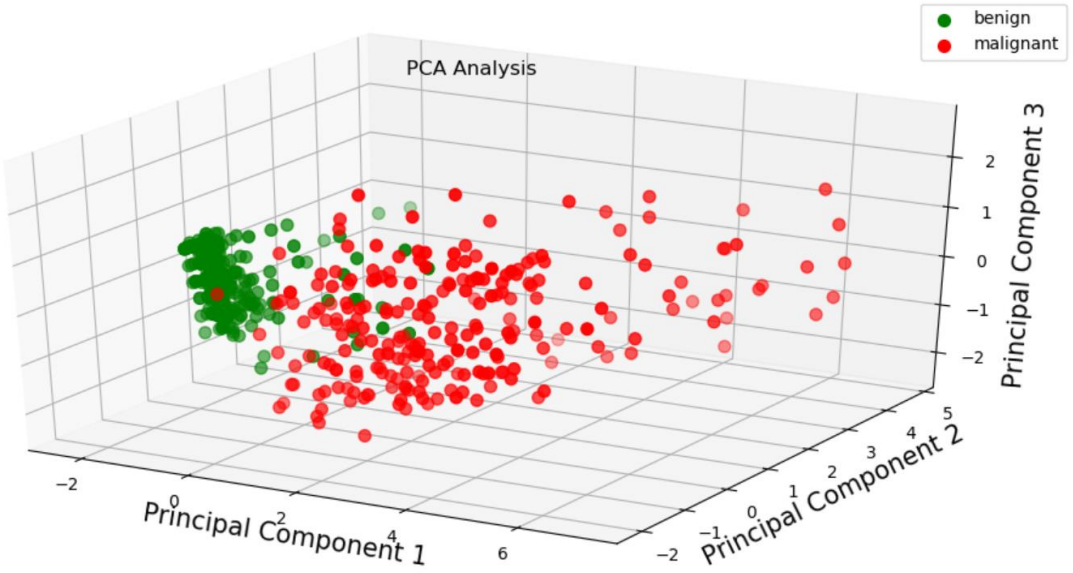
$$\textit{Precision} = \frac{TP}{TP + FP}$$

مقادیر به دست آمده بدون استفاده از PCA و با استفاده از آن :

```
bita@bita-K401UQK:~/programming/KNN-Classifier$ python KNN.py
Best K for prediction is : 3
Manual Accuracy: 0.943
Manual Recall: 0.939
Manual Precison: 0.902
bita@bita-K401UQK:~/programming/KNN-Classifier$ python KNN.py
Number of features reduced to : 3
Best K for prediction is : 5
Manual Accuracy: 0.950
Manual Recall: 0.959
Manual Precison: 0.904
bita@bita-K401UQK:~/programming/KNN-Classifier$
```

رس

ویژگی ها از ۶ عدد به ۳ عدد کاهش یافت و پیچیدگی محاسباتی کم شد.



۲. محاسبه ی احتمال پسین