Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Рубежный контроль №1
по дисциплине
«Методы машинного обучения»

Выполнил:
студент группы ИУ5-21М
Якубов А. Р.

____

Москва — 2019 г.

```
In [1]: # This Python 3 environment comes with many helpful analytics librarie
        # It is defined by the kaggle/python docker image: https://github.com/
        # For example, here's several helpful packages to load in

        import numpy as np # linear algebra
        import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

        # Input data files are available in the "../input/" directory.
        # For example, running this (by clicking run or pressing Shift+Enter)

        import os
        print(os.listdir("../input"))

        # Any results you write to the current directory are saved as output.

[]
```

```
In [2]: import pandas as pd
        import seaborn as sns
        from sklearn.datasets import load_iris
```

```
In [3]: sns.set(style="ticks", color_codes=True)
```

```
In [4]: iris = sns.load_dataset("iris")
```

```
In [5]: iris.head()
```

```
Out[5]:    sepal_length  sepal_width  petal_length  petal_width species
        0           5.1          3.5           1.4          0.2  setosa
        1           4.9          3.0           1.4          0.2  setosa
        2           4.7          3.2           1.3          0.2  setosa
        3           4.6          3.1           1.5          0.2  setosa
        4           5.0          3.6           1.4          0.2  setosa
```

```
In [6]: iris.describe()
```

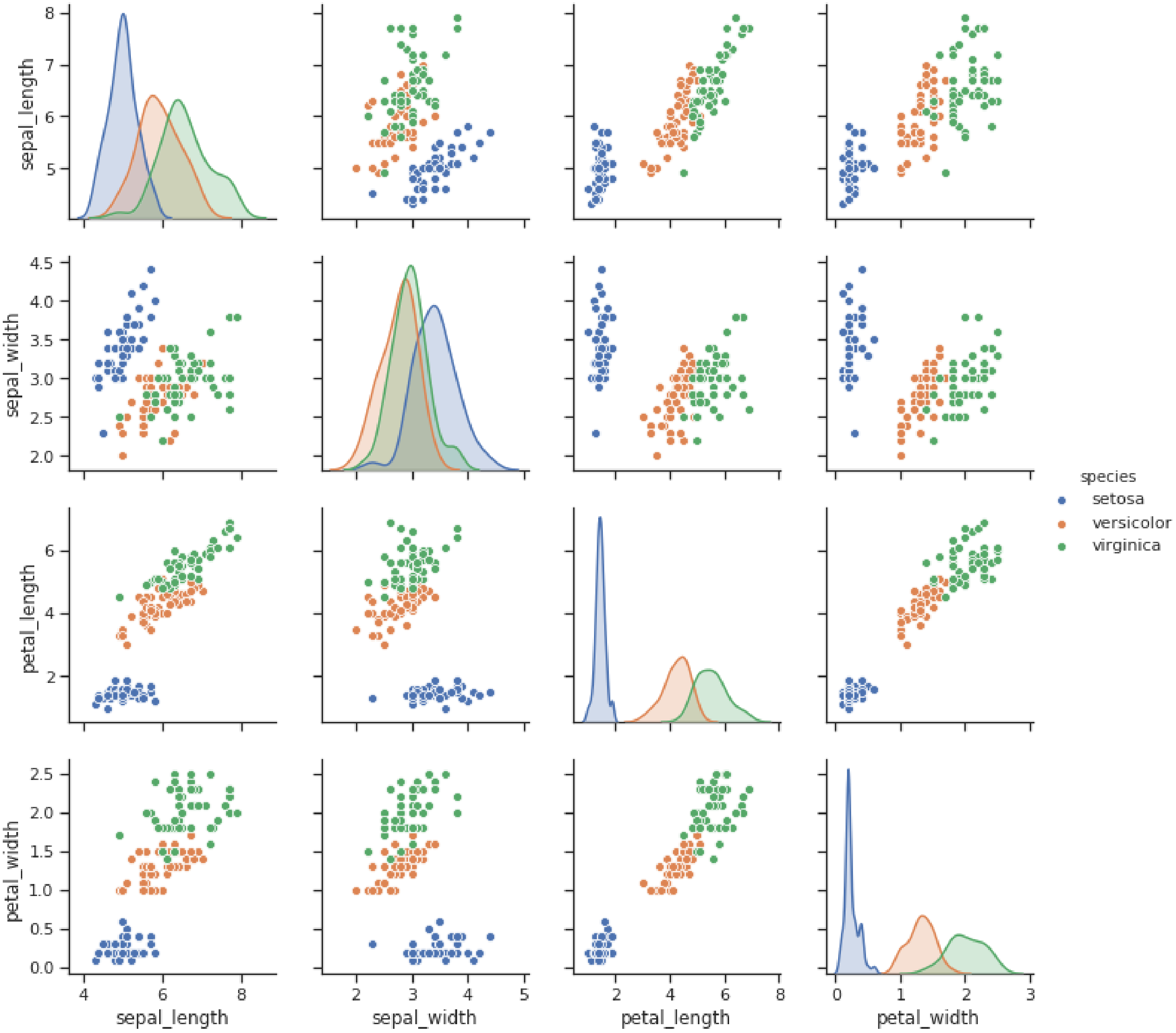
```
Out[6]:        sepal_length  sepal_width  petal_length  petal_width
        count    150.000000   150.000000    150.000000   150.000000
        mean       5.843333     3.057333      3.758000     1.199333
        std        0.828066     0.435866      1.765298     0.762238
        min        4.300000     2.000000      1.000000     0.100000
        25%        5.100000     2.800000      1.600000     0.300000
        50%        5.800000     3.000000      4.350000     1.300000
        75%        6.400000     3.300000      5.100000     1.800000
        max        7.900000     4.400000      6.900000     2.500000
```

```
In [7]: iris.species.describe()
```

```
Out[7]: count        150
        unique         3
        top       setosa
        freq          50
        Name: species, dtype: object
```

```
In [8]: g = sns.pairplot(iris, hue="species")

/opt/conda/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarnir
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



```
In [9]: iris.corr()

Out[9]:               sepal_length  sepal_width  petal_length  petal_width
        sepal_length      1.000000    -0.117570      0.871754     0.817941
        sepal_width      -0.117570     1.000000     -0.428440    -0.366126
        petal_length      0.871754    -0.428440      1.000000     0.962865
        petal_width       0.817941    -0.366126      0.962865     1.000000
```

```
In [10]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(iris.loc[:, :'pet
In [11]: from sklearn.svm import SVC
         from sklearn.metrics import accuracy_score
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.svm import SVC
```

```
In [12]: clf = SVC()
```

```
In [13]: clf.fit(X_train, y_train)
         y_pred_vote_clf = clf.predict(X_test)
         accuracy_score(y_test, y_pred_vote_clf)
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/svm/base.py:196: FutureWarning:
  "avoid this warning.", FutureWarning)
```

```
Out[13]: 1.0
```

```
In [14]: rnd_clf = RandomForestClassifier()
         rnd_clf.fit(X_train, y_train)
         y_pred_rf = rnd_clf.predict(X_test)
         accuracy_score(y_test, y_pred_rf)
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/ensemble/forest.py:246: FutureW
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
Out[14]: 1.0
```

```
In [15]: for feature_name, feature_importance in sorted(zip(iris.columns, rnd_
             print(feature_name, feature_importance)
```

```
petal_length 0.5647226607061763
petal_width 0.38320388152705237
sepal_length 0.0335199366741669
sepal_width 0.018553521092604497
```

```
In [16]:
```