Income Prediction（Kaggle exceed 50K）

Conclusion：

    Method          Accuracy   Parameter

1.  Random Forest   0.8548     RandomForestClassifier(n\_estimators=100, random\_state=0)

2.  AdaBoost        0.8693     AdaBoostClassifier(n\_estimators=100)

3.  XgBooost        0.8731     XGBClassifier(max\_depth=2, learning\_rate=1, n\_estimators=100,silent=True, objective='binary:logistic')

Note:

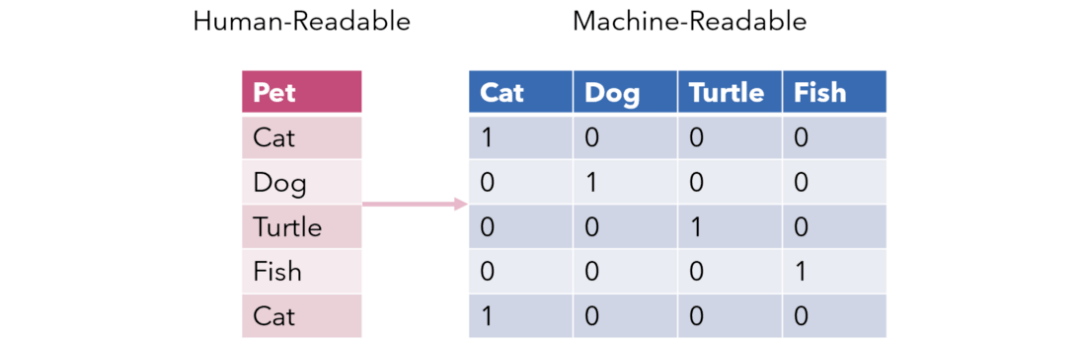
feature\_importance\_ is a tool for Rondom Forest.

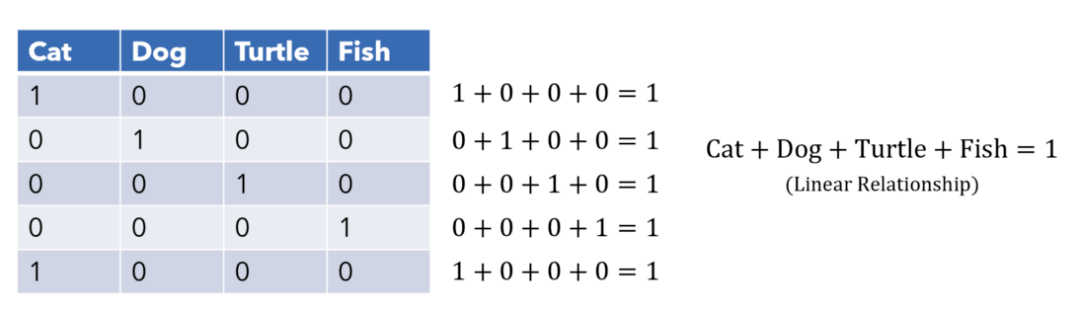
Random Forest drops the least important feature, which improves the accuracy.

HOwever,it is useless for AdaBoost and XgBooost to drop the least important feature.

one-hot encoding 是一种被广泛使用的编码方法，但也会造成维度过高等问题。

one-hot 编码（one-hot encoding）类似于虚拟变量（dummy variables），是一种将分类变量转换为几个二进制列的方法。其中 1 代表某个输入属于该类别。





# New knowledge

#1.

# Input data files are available in the "../input/" directory.

# For example, running this (by clicking run or pressing Shift+Enter)

# will list all files under the input directory

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

#2.

df.corr().style.format("{:.4}").background\_gradient(cmap=plt.get\_cmap('coolwarm'), axis=1)

#3.

for df2 in [X\_train, X\_test]:

    df2['workclass'].fillna(X\_train['workclass'].mode()[0], inplace=True)

#4.encode categorical variables with one-hot encoding

encoder = ce.OneHotEncoder(cols=['workclass', 'education', 'marital\_status', 'occupation',  'relationship', 'race', 'sex', 'native\_country'])

X\_train = encoder.fit\_transform(X\_train)

X\_test = encoder.transform(X\_test)

#4.

import seaborn as sns  #seaborn: statistical data visualization

#5.

import warnings

warnings.filterwarnings('ignore')

#6.

In Python missing data is represented by two values:

None: None is a Python singleton object that is often used for missing data in Python code.

NaN: NaN is an acronym for Not a Number. It is a special floating-point value recognized

by all systems that use the standard IEEE floating-point representation.

#7. Pandas isnull() and notnull() functions

Pandas offers two functions to test for missing values - isnull() and notnull().

df.isnull()

The above command checks whether each cell in a dataframe contains missing values or not.

If the cell contains missing value, it returns True otherwise it returns False.

df.isnull().sum()

The above command returns total number of missing values in each column in the dataframe.

df.isnull().sum().sum()

It returns total number of missing values in the dataframe.

df.isnull().mean()

It returns percentage of missing values in each column in the dataframe.

df.isnull().any()

It checks which column has null values and which has not.

The columns which has null values returns TRUE and FALSE otherwise.

df.isnull().any().any()

It returns a boolean value indicating whether the dataframe has missing values or not.

If dataframe contains missing values it returns TRUE and FALSE otherwise.

df.isnull().values.any()

It checks whether a particular column has missing values or not.

If the column contains missing values, then it returns TRUE otherwise FALSE.

df.isnull().values.sum()

It returns the total number of missing values in the dataframe.