Ушаков Владимир Александрович, группа МПиКИ

Лабораторная работа № 4

Знакомство с решением задачи регрессии

**Цель работы**

Решить задачу предсказания необходимости найма сотрудников.

**Код программы (внесённые изменения в шаблон кода выделены)**

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sb  
import numpy as np  
  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error  
from sklearn.ensemble import RandomForestRegressor  
from pandas.plotting import scatter\_matrix  
#%% md  
https://www.kaggle.com/datasets/helddata/turnover-team-data  
  
1 Team identifier (unique team number).  
  
2 Team size (number of people in the team).  
  
3 Team turnover 2014 (separation rate between 0 and 1).  
  
4 Country (1 = UK; 2 = United States; 3 = CANADA; 4 = SPAIN).  
  
5 SURVEY: ENGAGEMENT items COMBO (composite engagement percentage across the team).  
  
6 SURVEY: TeamLeader Rating (composite team leader percentage across the team).  
  
7 SURVEY: CSR rating (composite corporate social responsibility percentage across the team).  
  
8 SURVEY: Drive for Performance (composite percentage team score on perceived ‘Drive for performance’ percentage).  
  
9 SURVEY: Performance, Development and Reward (composite percentage team score on perceived fairness of performance, development and reward across the team).  
  
10 SURVEY: Work–Life Balance (composite percentage team score on perceived work–life balance across the team).  
  
11 UK dummy variable (0 = not UK; 1 = UK).  
  
12 USA dummy variable (0 = not United States; 1 = United States).  
  
13 Canada dummy variable (0 = not Canada; 1 = Canada).  
  
14 Spain dummy variable (0 = not Spain; 1 = Spain).  
#%%  
df = pd.read\_csv('data/Chapter\_6\_Turnover\_team\_DATA.csv')  
df = df.drop('TeamNumber', axis=1)  
df  
#%%  
df.info()  
#%%  
df.hist(figsize=(18, 18), xrot=-45);  
#%%  
plt.figure(figsize=(18, 18))  
sb.heatmap(df.corr(),  
 annot=True,  
 cmap='RdBu\_r',  
 vmin=-1,  
 vmax=1);  
#%%  
scatter\_matrix(df[["Engagement", "TeamLeader", "SociallyResponsible", "DriveForPerformance", "PerfDevReward", "WLB"]], alpha=0.5, figsize=(15, 15));  
#%%  
target = 'TeamSeparation'  
X = df.drop(target, axis=1)  
y = df[target]  
#%%  
ss = StandardScaler()  
#%%  
X\_n = ss.fit\_transform(X)  
X\_n  
#%%  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_n, y, train\_size=0.8, random\_state=42)  
#%%  
reg = LinearRegression().fit(X\_train, y\_train)  
reg.score(X\_train, y\_train)  
#%%  
y\_pred = reg.predict(X\_test)  
mean\_absolute\_error\_linear\_model = mean\_absolute\_error(y\_test, y\_pred)  
mean\_squared\_error\_linear\_model = mean\_squared\_error(y\_test, y\_pred)  
#%%  
print("MAE: {0:7.2f}, RMSE: {1:7.2f} для модели линейной регрессии".format(  
 mean\_absolute\_error\_linear\_model,  
 np.sqrt(mean\_squared\_error\_linear\_model)))  
#%%  
rfr = RandomForestRegressor().fit(X\_train, y\_train)  
rfr.score(X\_train, y\_train)  
#%%  
y\_pred = rfr.predict(X\_test)  
mean\_absolute\_error\_random\_forest\_model = mean\_absolute\_error(y\_test, y\_pred)  
mean\_squared\_error\_random\_forest\_model = mean\_squared\_error(y\_test, y\_pred)  
#%%  
print("MAE: {0:7.2f}, RMSE: {1:7.2f} для модели случайного леса".format(  
 mean\_absolute\_error\_random\_forest\_model,  
 np.sqrt(mean\_squared\_error\_random\_forest\_model)))  
#%%  
feature\_importance = pd.DataFrame(columns = ['Название признака', 'Важность признака'])  
#%%  
feature\_importance['Название признака'] = X.keys()  
feature\_importance['Важность признака'] = rfr.feature\_importances\_  
#%%  
# выведем полученную таблицу  
feature\_importance  
#%%  
feature\_importance.sort\_values(by='Важность признака', ascending=False)

**Результаты выполнения задания**

# В результате выполнения работы были получены следующие параметры на тестовых данных:

MAE: 0.13, RMSE: 0.17 для модели линейной регрессии

MAE: 0.12, RMSE: 0.16 для модели случайного леса

# Выводы

В результате выполнения работы была успешно решена задача регрессии