МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э.Баумана

Отчет по рубежному контролю №2
по курсу «Технологии машинного обучения»

Методы построения моделей машинного обучения.

Подготовил
Ионов С.А.
ИУ5-62Б
Вариант №10

1) Описание задания

Для заданного набора данных (по Вашему варианту) постройте модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей используйте методы 1 и 2 (по варианту для Вашей группы). Оцените качество моделей на основе подходящих метрик качества (не менее двух метрик). Какие метрики качества Вы использовали и почему? Какие выводы Вы можете сделать о качестве построенных моделей? Для построения моделей необходимо выполнить требуемую предобработку данных: заполнение пропусков, кодирование категориальных признаков, и т.д.

2) Текст программы

0. Подготовка

```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import balanced_accuracy_score, plot_roc_curve, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

# отбираем 5000 строк из всего датасета
data = pd.read_csv('data/hotel_bookings.csv', nrows=5000)

# Оцениваем баланс классов целевого признака
data['is_canceled'].value_counts()/data['is_canceled'].shape[0]*100

# Проверяем процент пропусков в данных для всех колонок
(data.isnull().sum()/data.shape[0]*100).sort_values(ascending=False)

# Строим гистограмму распределения для импутируемого признака
g = sns.kdeplot(data=data, x="agent", shade=True)
g.set_xlabel("agent", size = 15)
g.set_ylabel("Frequency", size = 15)
plt.title('Distribution of agent', size = 18)

data.drop(['company'], axis=1, inplace=True)
data.dropna(subset=['country'], axis=0, inplace=True)
indicator = MissingIndicator()
```

```
mask_missing_values_only = indicator.fit_transform(data[['agent']])
imp_num = SimpleImputer(strategy='median')
data_num_imp = imp_num.fit_transform(data[['agent']])
data['agent'] = data_num_imp
filled_data = data_num_imp[mask_missing_values_only]
print('agent', 'median', filled_data.size, filled_data[0], filled_data[filled_data.size-1], sep='; ')

# Проверяем, что импутация не разрушила распределение
g = sns.kdeplot(data=data, x="agent", shade=True)
g.set_xlabel("agent", size = 15)
g.set_ylabel("Frequency", size = 15)
plt.title('Distribution of agent', size = 18)

# Проверяем категориальные признаки на уникальность
col_obj = data.dtypes[data.dtypes==object].index.values.tolist()
for i in enumerate(col_obj):
    uniq_obj = data[i[1]].unique()
    print(f'{i[0]+1}. {i[1]}: {uniq_obj} | КОЛ-ВО: {len(uniq_obj)}')

# Копируем датасет и применяем label-encoding категориальных признаков для
составления корреляционной матрицы
# и последующего применения в модели Random Forest
dataLE = data.copy()
le = LabelEncoder()
col_obj = dataLE.dtypes[dataLE.dtypes==object].index.values.tolist()
for i in col_obj:
    dataLE[i] = le.fit_transform(dataLE[i])

plt.figure(figsize=(10,10))
g = sns.heatmap(dataLE.corr())

# Оцениваем важность признаков для целевого
(dataLE.corr()['is_canceled']*100).sort_values(ascending=False)

del_data = (dataLE.corr()['is_canceled']*100).sort_values(ascending=False)
del_col = del_data[(del_data < 10) & (del_data > -10) | (del_data.isnull())].index.values.tolist()
data.drop(columns=del_col, inplace=True)
dataLE.drop(columns=del_col, inplace=True)

# Выполняем one-hot encoding и масштабирование для применения в SVM
col_num = data.dtypes[data.dtypes!=object].index.values.tolist()
col_num.remove('is_canceled')
se = StandardScaler()
data[col_num] = se.fit_transform(data[col_num])
data = pd.get_dummies(data, drop_first=True)

TEST_SIZE = 0.3
RANDOM_STATE = 0
```

```
dataLE_X = dataLE.drop(columns='is_canceled')
dataLE_y = dataLE['is_canceled']
data_X = data.drop(columns='is_canceled')
data_y = data['is_canceled']
dataLE_X_train, dataLE_X_test, dataLE_y_train, dataLE_y_test = train_test_split(dataLE_X,
dataLE_y, \
                                                                test_size = TEST_SIZE, \
                                                                random_state= RANDOM_STATE)
data_X_train, data_X_test, data_y_train, data_y_test = train_test_split(data_X, data_y, \
                                                        test_size = TEST_SIZE, \
                                                        random_state= RANDOM_STATE)

def print_metrics(X_train, Y_train, X_test, Y_test, clf):
    clf.fit(X_train, Y_train)
    target = clf.predict(X_test)
    print(f'Сбалансированная оценка: {balanced_accuracy_score(Y_test, target)}')
    fig, ax = plt.subplots()
    plot_roc_curve(clf, X_test, Y_test, ax=ax)
    ax.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',
            label='Chance', alpha=.8)
    ax.set(xlim=[-0.05, 1.05], ylim=[-0.05, 1.05],
           title="Receiver operating characteristic")
    ax.legend(loc="lower right")
    plt.show()
    print(f'Матрица ошибок:\n {confusion_matrix(Y_test, target)}')
```

1. SVM

```
print_metrics(data_X_train, data_y_train, data_X_test, data_y_test,
SVC(random_state=RANDOM_STATE))
```

2. Random Forest

```
print_metrics(dataLE_X_train, dataLE_y_train, dataLE_X_test, dataLE_y_test,
RandomForestClassifier(random_state=RANDOM_STATE))
```

3) Экранные формы с примерами выполнения программы

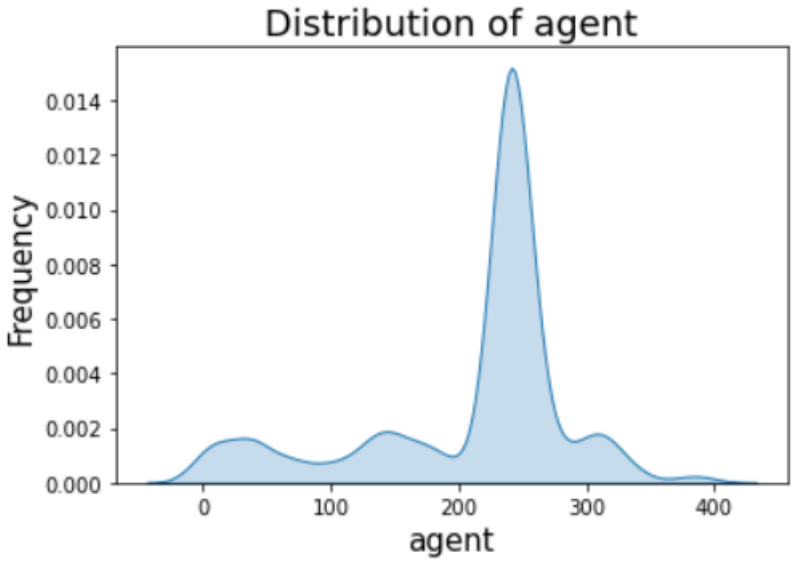
- Для рубежного контроля №2 согласно варианту взят следующий датасет. Будем решать задачу бинарной классификации: будет ли отменено бронирование данной комнаты в отеле (is_canceled - целевой признак).
- Классы целевого признака не настолько сильно дисбалансированны, чтобы применять стратегии сэмплинга для выравнивания баланса:

```
0    54.04
1    45.96
```

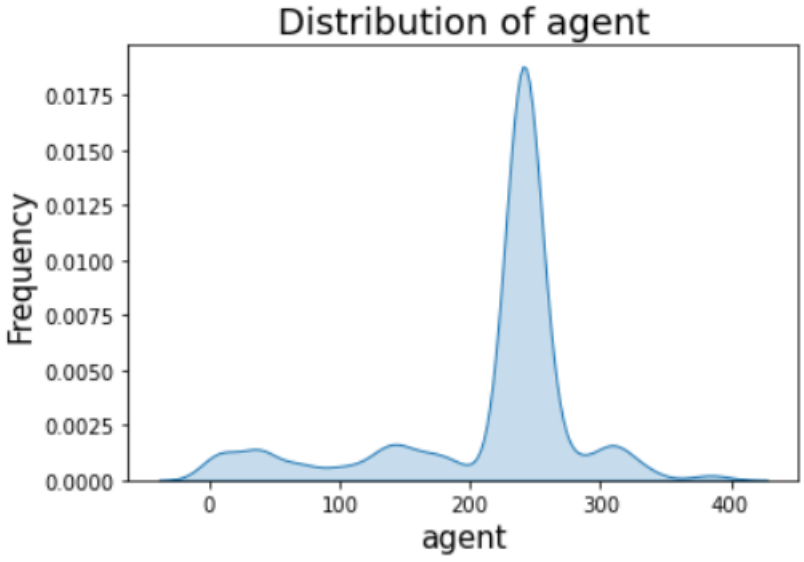
- В датасете присутствуют 3 колонки с пропусками. Столбец company содержит много пропущенных данных, поэтому данный столбец будет удален. Столбец agent содержит приемлемый процент пропусков для восстановления. Строки, для которых в столбце country содержатся пропуски, будут удалены:

```
company                         94.16
agent                           16.28
country                          0.04
lead_time                        0.00
arrival_date_year                0.00
arrival_date_month               0.00
```

- Распределение столбца agent мультимодально, поэтому для импутации будем использовать медиану:



- После применения импутации распределение практически не изменилось

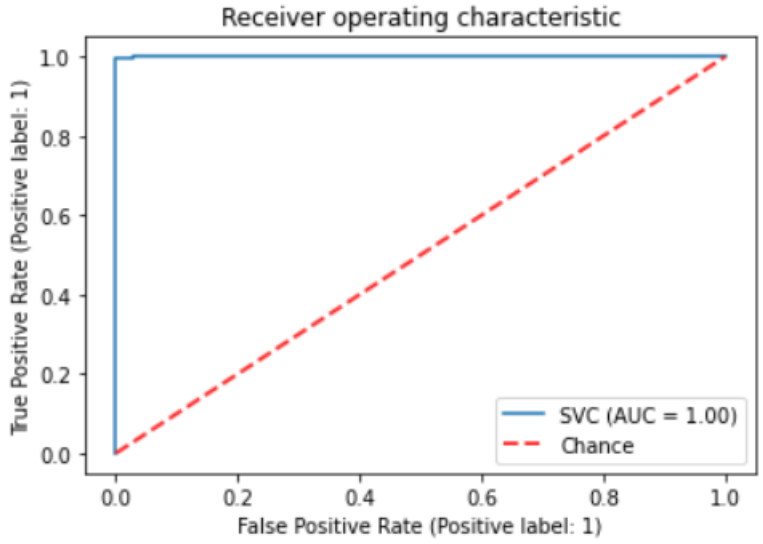


- Оцениваем важность признаков по отношению к целевому признаку на основе корреляционной матрицы. Столбцы, коэффициент корреляции которых по модулю меньше, чем 10, или NaN (ввиду однозначности значения), будут удалены:

```
is_canceled                     100.000000
country                          52.533878
arrival_date_year                29.437152
deposit_type                     19.751308
lead_time                         7.588779
market_segment                    5.883349
distribution_channel              4.700574
adults                            4.537695
stays_in_weekend_nights           2.942242
children                          2.469151
stays_in_week_nights              0.049425
reservation_status_date          -0.040024
customer_type                    -0.979502
meal                             -1.987424
reserved_room_type               -2.664975
babies                           -2.954529
agent                            -3.553828
arrival_date_day_of_month        -3.558175
adr                              -4.973463
total_of_special_requests        -8.264548
days_in_waiting_list            -11.344538
arrival_date_month              -16.216285
booking_changes                 -18.118893
assigned_room_type              -19.255699
arrival_date_week_number        -24.489474
required_car_parking_spaces     -29.537194
reservation_status              -87.450209
hotel                                  NaN
is_repeated_guest                      NaN
previous_cancellations                 NaN
previous_bookings_not_canceled         NaN
Name: is_canceled, dtype: float64
```

- Оценки для SVM:

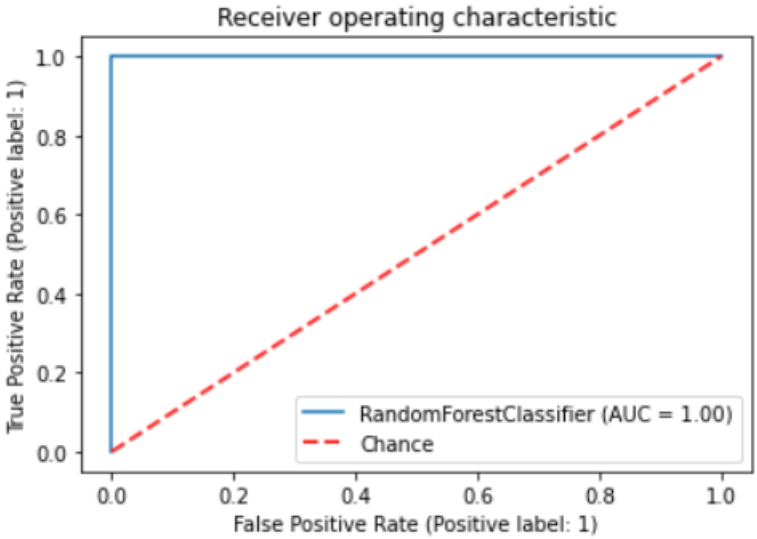
```
Сбалансированная оценка: 0.9985141158989599
```




```
Матрица ошибок:
 [[827   0]
 [  2 671]]
```

- Оценки для Random Forest:

```
Сбалансированная оценка: 1.0
```




```
Матрица ошибок:
 [[827   0]
 [  0 673]]
```

- Выводы, ответы на вопросы к РК:

В данной работе для оценки моделей были использованы следующие метрики, подходящие для задачи бинарной классификации: balanced accuracy, так как данная метрика хорошо интерпретируется и используется при несбалансированных классах; ROC-кривая (AUC), так как позволяет по графику понять, насколько модель может минимизировать FP (False Positive), т.е. признавать отмененным заказ, который таковым не является, и минимизировать FN (False Negative), т.е. признавать бронированным заказ, который был отменен; confusion matrix, так как, хотя и метрикой в полной мере не является, позволяет увидеть общую картину по всем видам ошибок.

По результатам оценивания можно сделать следующий вывод: модель Random Forest обладает немного большей предсказательной способностью, чем Support Vector Machine. Но при этом обе модели могут использоваться для предсказания, будет ли заказ по бронированию отменен, с минимальным количеством ошибок.