Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г. В. Плеханова»

Кафедра информатики

Выпускная квалификационная работа

по программе профессиональной переподготовки «Процедурно-ориентированное программирование в прикладных задачах анализа данных в экономике»

на тему

«Прогнозирование показателей миграции в РФ с использованием моделей машинного обучения»

Выполнил:
Семенов Роман Валерьевич

Преподаватель:
ст. преп. Савинова Виктория Михайловна

Москва
2022

1. Понимание бизнес-целей

1.1. Понимание бизнеса

Перед аналитическим отделом группы компаний работающей в сфере инвестиционной иммиграции в страны ЕС и Карибского бассейна, а также предоставляющей услуги по программе EB-5 в США, в связи с запланированным открытием департамента релокации в Москве для работы с иностранными гражданами (экспатами), руководством поставлена задача спрогнозировать число прибывших на территорию РФ с 1 кв. 2021 года по 4 кв. 2021 года включительно.

Настоящий анализ и его интерпретация будут включены в комплекс стратегического планирования и бюджетирования группы компаний.

Проект призван оптимизировать расходы на департамент релокации и предстоящий комплекс маркетинга при выводе на рынок новой услуги.

1.2. Доступные ресурсы

Для успешной реализации настоящего проекта необходимы следующие категории специалистов: аналитик данных, бизнес-аналитик, руководитель проекта.

Аналитический отдел располагает всем необходимым оборудованием для проведения анализа данных.

1.3. Риски

1. Несоблюдение сроков проекта
2. Риск нехватки и неполноты данных
3. Глобальные и/или локальные экономические кризисы

1.4. Ограничения

Ограничение сроков: 5 рабочих дней.

Ставки по сотрудникам:

- аналитик данных – 1 ставка;
- бизнес-аналитик – 1 ставка;
- руководитель проекта — 1 ставка.

1.5. Цели исследования данных

Задачами анализа данных, в рамках настоящего проекта являются:

1. Построение визуализации корреляции общего исследуемого набора данных: KR, M2, FW, REZ, VVP, MIGRAIN (см. п.п. 2.1.1.)
2. Для наилучшей корреляции выполняется построение диаграммы рассеивания с вектором найденных коэффициентов m и b в модели линейной регрессии.
3. Прогнозирование числа прибывших - MIGRAIN (см. п.п. 2.1.1.) с использованием моделей регрессии. В качестве моделей будут рассмотрены следующие методы:
   - линейная регрессия (LinearRegression);

- линейная регрессия методом ближайшего соседа (KNeighborsRegressor);
- дерево решений (DecisionTreeRegressor);
- случайный лес (RandomForestRegressor).

1.6. Критерии успешности изучения данных.

Метрики оценки точности и качества построенных моделей:

Для моделей регрессии качество модели определяется с использованием коэффициента детерминации (R^2), а точность модели определяется на основании средней относительной ошибки (MAPE).

Границы значений метрик:

- R^2 должен быть больше либо равен 0.8;

- MAPE не более 10 %.

2. Начальное изучение данных.

2.1. Сбор данных.

2.1.1. Для прогнозирования используются внешние данные:

1. Пять сценарных (прогнозных) показателей ЦБРФ с 1 кв. 2021 года по 4 кв. 2021 года включительно и пять фактических показателей за период с 1 кв. 2013 года по 4 кв. 2020 года включительно :

KR - Ключевая ставка ЦБ РФ (% годовых);

M2 - Темп прироста денежной массы в национальном распределении (млрд. руб.);

FW - Цена на нефть марки Urals, средняя за год (долл. США за баррель);

REZ - Изменение международных резервов РФ (% к предыдущему году);

VVP - Валовый внутренний продукт (млрд. рублей).

2. Данные Росстата по прибывшим за период с 1 кв. 2013 года по 4 кв. 2020 года включительно:

MIGRAIN - Число прибывших (человек).

2.2. Описание данных

Объем данных – 13.0 КиБ

Типы, виды данных и схемы кодирования представлены в таблице:

Показатели	Тип данных	Вид данных	Схема кодирования
KR	Число с плавающей запятой (float)	Непрерывный	-
M2	Число с плавающей запятой (float)	Непрерывный	-

FW	Число с плавающей запятой (float)	Непрерывный	-
REZ	Число с плавающей запятой (float)	Непрерывный	-
VVP	Число с плавающей запятой (float)	Непрерывный	-
MIGRAIN	Число с плавающей запятой (float)	Непрерывный	-

Формат данных – файл csv, разделитель – “;”.

2.3. Исследование данных

Построение описательной статистики:

	MIGRAIN	KR	M2	FW	REZ	VVP
count	32.000000	36.000000	36.000000	36.000000	36.000000	36.000000
mean	146446.093750	7.752778	40642.556944	64.470556	1.976667	23574.269167
std	22189.603598	2.532530	10240.823613	23.407679	18.731181	4078.997125
min	92193.000000	4.250000	26760.570000	32.800000	-68.780000	16370.000000
25%	132072.500000	5.500000	31459.395000	47.645000	-7.775000	20365.260000
50%	149871.000000	7.500000	38813.970000	60.805000	5.575000	23184.010000
75%	160765.000000	9.812500	46685.145000	70.815000	12.670000	27616.140000
max	210350.000000	14.000000	58892.700000	110.900000	30.560000	29815.600000

Проверка пропущенных значений:

‘’’

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 36 entries, 0 to 35

Data columns (total 6 columns):

#	Column	Non-Null Count	Dtype
---	------	--------------	-----

0	MIGRAIN	32 non-null	float64
1	KR	36 non-null	float64
2	M2	36 non-null	float64
3	FW	36 non-null	float64
4	REZ	36 non-null	float64
5	VVP	36 non-null	float64

dtypes: float64(6)

memory usage: 1.8 KB

None

‘’’

В исследуемом диапазоне пропущенные значения отсутствуют.

3. Подготовка данных

Данные не требуют очистки, так как не выявлено дубликатов, пропусков, противоречий и экстремальных значений.

Осуществлено разбиение на обучающую, тестовую и прогнозную выборки:

X = df[['VVP', 'KR', 'M2', 'FW', 'REZ']]

y = df['MIGRAIN']

x_train = X.iloc()[:28] — обучающая выборка

x_test = X.iloc()[28:32] — тестовая выборка

y_train = y.iloc()[:28] — обучающая выборка

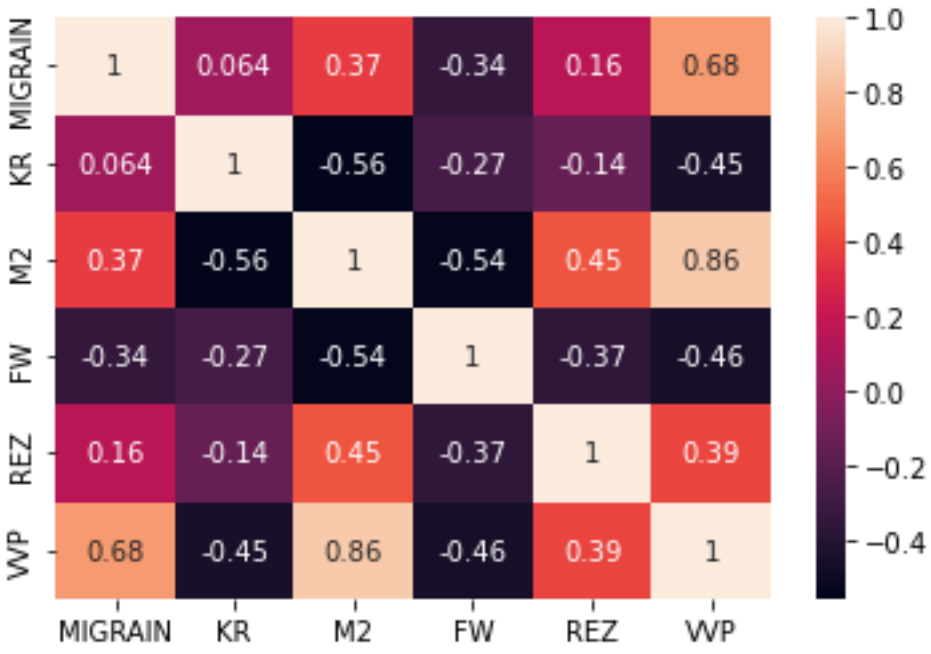
y_test = y.iloc()[28:32] — тестовая выборка

x_val = X.iloc()[32:] — прогнозная выборка

Для достижения наилучшего результата разбиение обучающей и тестовой выборок произведено в пропорции ~ 90/10 (обучающая выборка должна быть больше тестовой).

4. Моделирование и оценка результатов

4.1. Корреляционная матрица с применением библиотеки Seaborn.



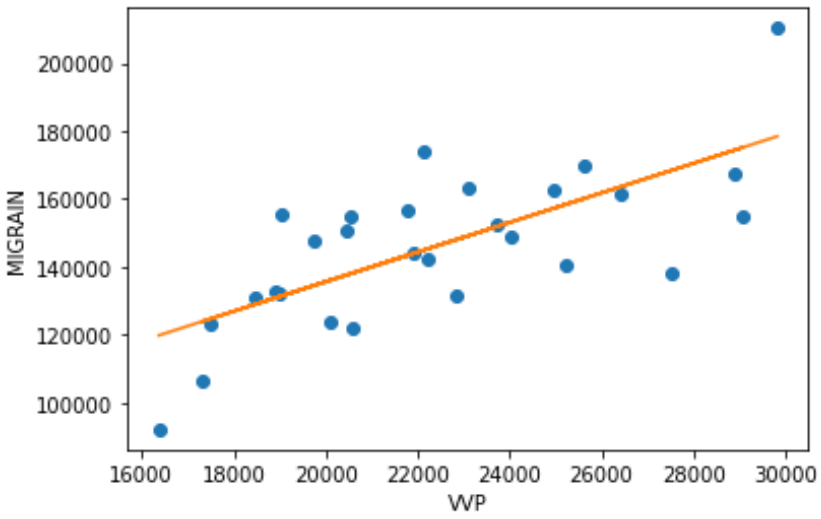
Из матрицы следует, что лучшая корреляция числа прибывших – MIGRAIN, наблюдается с Внутренним Валовым Продуктом – VVP и составляет 0.68

4.2. Модель линейной регрессии

===LinearRegression===

Коэффициент b - сдвиг прямой из уравнения y = m*x + b: 48548.60979889624

Коэффициент m - наклон прямой из уравнения y = m*x + b: [4.35682775]



Приведенная выше диаграмма рассеивания и найденный в процессе обучения вектор демонстрируют качество обучения модели.

- Оценка качества обучения модели R^2: 0.5024393411478226

- Точность модели MAPE (отклонение от факта): 0.11503736884929057

- Прогнозирование MIGRAIN на заданный период:

[174033.52703482 170268.79217415 174358.54638514 176886.37784696]

4.3. Линейная регрессия методом ближайшего соседа

===KNeighborsRegressor===

- Параметры модели:

KNeighborsRegressor(n_neighbors=5)

- Оценка качества обучения модели R^2: 0.49567071356285186

- Точность модели MAPE (отклонение от факта): 0.11044576576475315

- Прогнозирование MIGRAIN на заданный период:

[171278.2 171278.2 171278.2 171278.2]

4.4. Дерево решений

===DecisionTreeRegressor===

- Параметры модели:

DecisionTreeRegressor(min_samples_split = 3, min_samples_leaf = 3)

- Оценка качества обучения модели R^2: 0.7201486041375744

- Точность модели MAPE (отклонение от факта): 0.1700901036913446

- Прогнозирование MIGRAIN на заданный период:

[171278.2 171278.2 171278.2 171278.2]

4.5. Случайный лес

===RandomForestRegressor===

- Параметры модели:

RandomForestRegressor(min_samples_split = 3, min_samples_leaf = 3, n_estimators = 16)

- Оценка качества обучения модели R^2: 0.6463375852889883

- Точность модели MAPE (отклонение от факта): 0.08800950391803786

- Прогнозирование MIGRAIN на заданный период:

[159937.59558532 153829.9609127 156214.62237103 156214.62237103]

Корреляционная связь анализируемых данных имеется, но недостаточна для более эффективного прогноза. Наилучшие показатели точности и качества показала модель дерева решений. Однако, не было достигнуто необходимое значение коэффициента детерминации R^2. Принятие результатов модели при формировании бюджета зависит от итогов совещания руководства.

5. Внедрение.

Модель дерева решений имеет наиболее высокий результат. В дальнейшем возможно использование данной модели на большем и/или не сгруппированном наборе данных по кварталам, как в настоящем проекте, что в свою очередь даст лучший результат. Внедрение модели зависит от итогов совещания руководства.

Приложение 1. Код программы Python для проведенного анализа

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep  9 18:02:30 2022

@author: Roman Semenov
"""

'''Прогнозирование показателей миграции в РФ с использованием моделей машинного обучения'''



import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_percentage_error
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor

pd.set_option('display.max_columns', None)

def lin_reg (x_train, x_test, y_train, y_test, y_val):
    reg = LinearRegression()
    reg.fit(x_train, y_train)
    y_pred = reg.predict(x_train)
    y_pred2 = reg.predict(x_test)
    y_pred3 = reg.predict(x_val)
    print('===LinearRegression===')
    print('Коэффициент b - сдвиг прямой из уровнения y = m*x + b:', reg.intercept_)
    print('Коэффициент m  - наклон прямой из уровнения y = m*x + b:', reg.coef_)
```

```
    print('----------------------')
    print('Оценка качества обучения модели R^2:', r2_score (y_train, y_pred))
    print('Точность модели MAPE (отклонение от факта):', mean_absolute_percentage_error(y_test, y_pred2))
    print('Прогнозирование MIGRAIN на заданный период:', y_pred3)
    print()
    plt.plot(x_train, y_train, 'o')
    plt.plot(x_train, y_pred)
    plt.xlabel('VVP')
    plt.ylabel('MIGRAIN')
    plt.show()
    return y_pred


def neighbor (x_train, x_test, y_train, y_test, n, y_val):
    reg = KNeighborsRegressor(n_neighbors=n)
    reg.fit(x_train, y_train)
    y_pred = reg.predict(x_train)
    y_pred2 = reg.predict(x_test)
    y_pred3 = reg.predict(x_val)
    print('===KNeighborsRegressor===')
    print('Оценка качества обучения модели R^2:', r2_score(y_train, y_pred))
    print('Точность модели MAPE (отклонение от факта):', mean_absolute_percentage_error(y_test, y_pred2))
    print('Прогнозирование MIGRAIN на заданный период:', y_pred3)
    print()
    return y_pred


def des_tree (x_train, x_test, y_train, y_test, y_val):
    reg = DecisionTreeRegressor(min_samples_split = 3, min_samples_leaf = 3)
    reg.fit(x_train, y_train)
```

```
    y_pred = reg.predict(x_train)
    y_pred2 = reg.predict(x_test)
    y_pred3 = reg.predict(x_val)
    print('===DecisionTreeRegressor===')
    print('Оценка качества обучения модели R^2:', r2_score(y_train, y_pred))
    print('Точность модели MAPE (отклонение от факта):', mean_absolute_percentage_error(y_test, y_pred2))
    print('Прогнозирование MIGRAIN на заданный период:', y_pred3)
    print()
    return y_pred

def RFR (x_train, x_test, y_train, y_test, y_val):
    reg = RandomForestRegressor(min_samples_split = 3, min_samples_leaf = 3, n_estimators = 16)
    reg.fit(x_train, y_train)
    y_pred = reg.predict(x_train)
    y_pred2 = reg.predict(x_test)
    y_pred3 = reg.predict(x_val)
    print('===RandomForestRegressor===')
    print('Оценка качества обучения модели R^2:', r2_score(y_train, y_pred))
    print('Точность модели MAPE (отклонение от факта):', mean_absolute_percentage_error(y_test, y_pred2))
    print('Прогнозирование MIGRAIN на заданный период:', y_pred3)
    print()
    return y_pred

df = pd.read_csv('socio-economic_data.csv', sep = ';', encoding = 'cp1251')
df_stat = df[['MIGRAIN', 'KR', 'M2', 'FW', 'REZ', 'VVP']]
print(df_stat.describe())
print(df_stat.info())

df = df[['MIGRAIN', 'KR', 'M2', 'FW', 'REZ', 'VVP']]
```

```
X = df[['VVP']]
y = df['MIGRAIN']

x_train = X.iloc()[:28]
x_test = X.iloc()[28:32]

y_train = y.iloc()[:28]
y_test = y.iloc()[28:32]

x_val = X.iloc()[32:]

y_liner = lin_reg(x_train, x_test, y_train, y_test, x_val)

X = df[['VVP', 'KR', 'M2', 'FW', 'REZ']]
y = df['MIGRAIN']

x_train = X.iloc()[:28]
x_test = X.iloc()[28:32]

y_train = y.iloc()[:28]
y_test = y.iloc()[28:32]

x_val = X.iloc()[32:]

y_neighbor = neighbor (x_train, x_test, y_train, y_test, 5, x_val)
y_des_tree = des_tree (x_train, x_test, y_train, y_test, x_val)
y_RR = RFR (x_train, x_test, y_train, y_test, x_val)

sns.heatmap(df.corr(), annot = True)
```