УДК 681.5

ПОСТРОЕНИЕ МОДЕЛЕЙ И ПРОГНОЗИРОВАНИЕ С ПРИМЕНЕНИЕМ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ЗАДАЧ СЕЛЬСКОГО ХОЗЯЙСТВА

Сабитов[1] Ч.Б., Алмасбекова[1] З., Орозобекова[2] А.К., Сабитов[1] Б.Р.
[1]Кыргызский национальный университет им. Баласагына,
[2]КГУСТА им. Н. Исанова

В данной статье исследуется процесс построения моделей на основе алгоритмов машинного обучения. В последние годы в зависимости от применения различных удобрений (калий, фосфор, азот и др.) , погодных условий (температура ,влажность и др.) , а также ухудшения плодородия почв сохранение урожайности от сельскохозяйственных культур для многих фермеров является первостепенной задачей. Получены результаты анализа различных алгоритмов

Ключевые слова. Машинное обучение, алгоритмы, база данных, сельскохозяйственные культуры, прогнозирование, тестирование

АЙЫЛ ЧАРБА МАСЕЛЕЛЕРИ БОЮНЧА МАШИНА ҮЙРӨНҮҮ АЛГОРИТМДЕРИН КОЛДОНУУ МЕНЕН МОДЕЛДИ ТҮЗҮҮ ЖАНА БОЛЖОНДОО

Сабитов[1] Ч.Б., Алмасбекова[1] З., Орозобекова[2] А.К., Сабитов[1] Б.Р.
[1] Баласагын атын. Кыргыз улуттук университети
[2]Н.Исанов атын. КМКТАУ

Бул макалада машина үйрөнүү алгоритмдеринин негизинде моделдерди куруу процесси изилденет. Акыркы жылдары ар кандай жер семирткичтерди (калий, фосфор, азот ж.б.) колдонууга, аба ырайынын шарттарына (температура, нымдуулук ж.б.), ошондой эле кыртыштын асылдуулугунун начарлашына жараша көптөгөн дыйкандар үчүн айыл чарба өсүмдүктөрүнүн түшүмдүүлүгүн сактап калуу башкы милдет болуп эсептелнет. Ар кандай алгоритмдердин анализинин натыйжалары алынган.

Баштапкы сөздөр: Машиналарды үйрөнүү, алгоритмдер, маалымат базасы, өсүмдүктөр, болжолдоо, тестирлөө.

MODEL BUILDING AND FORECASTING USING MACHINE LEARNING ALGORITHMS FOR AGRICULTURAL PROBLEMS

Sabitov[1] Ch.B., Almasbekova[1] Z., Orozobekova[2] A.K., Sabitov[1] B.R.
[1]Kyrgyz National University named of Balasagyn,
[2]KSUCTA named of N. Isanova

This article explores the process of building models based on machine learning algorithms. In recent years, depending on the use of various fertilizers (potassium, phosphorus, nitrogen, etc.), weather conditions (temperature, humidity, etc.), as well as deterioration of soil fertility, the preservation of crop yields for many farmers is a paramount task. The results of the analysis of various algorithms are obtained.

Keywords. Machine learning, algorithms, database, crops, forecasting, testing

Содержание

Во многих случаях прибыли от урожайности многих фермерских полей, в отдельно взятые годы, не покрывают даже затраты, которые внесли фермеры на выращивания того или иного вида сельскохозяйственных культур. Наблюдается, также процесс оттока фермеров, которые сильно влияет на выполнения продовольственной программы. Внедрение современных технологий орошения, научные подходы на изучения процессов улучшения урожайности, необходимы для сельского хозяйства. С этой целью, для уменьшения потерь урожайности в данной статье строится различные модели улучшения урожайности с применением машинного обучения. Для построения моделей с помощью машинного обучения мы будем использовать основные библиотеки из пакета Python.

Метод исследования

Сначала изучим базу данных и основных влияющих факторов на урожайность для регионов Кыргызской республики. Данная база данных содержит различные данные о внесенных удобрений, и климатических особенностей региона температуры, влажность и осадки. Особую роль играет изучение кислотности почв, которая определяет основу получения урожайности от различных культур. База данных охватывает, почти все

растения, которые выращивают земледельцы регионов нашей страны. Вот собранная база данных локальным расположением и с расширением .csv df = pd.read_csv('C:/Users/User/Desktop/A_notebooks_Urojai/Data-processed/crop_recommendation_work.csv')

Отобразим данные. Первые пять строк базы данных, которые выглядят следующим образом, например картошки

```
In [3]: df.head()
```

Out[3]:

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	potato
1	85	58	41	21.770462	80.319644	7.038096	226.655537	potato
2	60	55	44	23.004459	82.320763	7.840207	263.964248	potato
3	74	35	40	26.491096	80.158363	6.980401	242.864034	potato
4	78	42	42	20.130175	81.604873	7.628473	262.717340	potato

Для груши выборка выглядит так

```
In [4]: df.tail()
```

Out[4]:

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	pear
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	pear
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	pear
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	pear
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	pear

Всего собранных по регионам записей в базе данных

df.size

17600

Количество строк и столбцов датафрейма

df.shape

(2200, 8)

Выведем теперь название основных столбцов базы данных, влияющих на урожайность сельхозрастения

df.columns

Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')

Посмотрим на уникальные метки сельскохозяйственных культур. Это обычно название растения. Мы выбрали следующие наиболее часто выращиваемые культуры фермерами нашей Республики :

'potato'-картофель, 'maize'-кукуруза, 'wheat'-пшеница, 'kidneybeans'-фасоль, 'pigeonpeas'-горох, 'mothbeans'-бобы, 'mungbean'-маш,'black currant'-черная смородина, 'lentil'-чечевидца, 'pomegranate'-гранат, 'barley'-ячмень,'apricot'-абрикос, 'grapes'-виноград, 'watermelon'-арбуз, 'muskmelon'-дыня, 'apple'-яблоко,'cherry'-вишня, 'cucumber'-огурец, 'tomato'-помидор, 'cotton'-хлопок, 'plum'-слива, 'pear'-груша

Вот уникальные метки набора данных

df['label'].unique()

array(['potato', 'maize', 'wheat', 'kidneybeans', 'pigeonpeas',

'mothbeans', 'mungbean', 'black currant', 'lentil', 'pomegranate',

'barley', 'apricot', 'grapes', 'watermelon', 'muskmelon', 'apple',

'cherry', 'cucumber', 'tomato', 'cotton', 'plum', 'pear'],

dtype=object)

Опишем теперь типы данных признаков входящих в базу данных

df.dtypes

N int64 P int64 K int64 temperature float64 humidity float64

ph float64 rainfall float64 label object

dtype: object

Выведем теперь количество записей растений входящих в данную базу данных т.е. у нас имеются следующие растения с названиями меток и количеством данных.

df['label'].value_counts()

apple 100 cherry 100 muskmelon 100 barley 100 watermelon 100

pigeonpeas 100 cotton 100 mothbeans 100 mungbean 100 tomato 100

kidneybeans 100 cucumber 100 lentil 100 grapes 100 plum 100

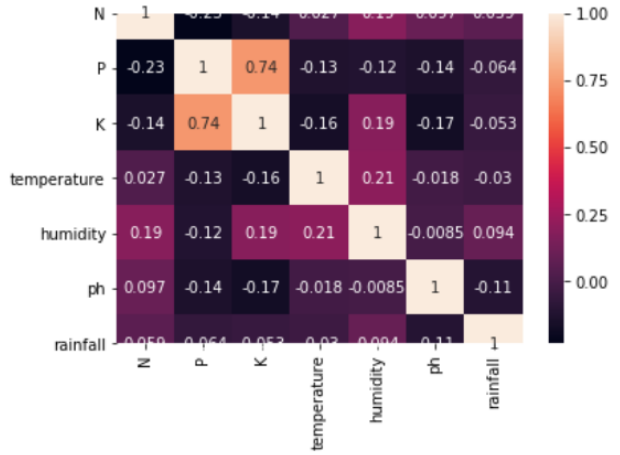
wheat 100 pomegranate 100 apricot 100 pear 100 black currant 100

potato 100 maize 100

Name: label, dtype: int64

Используя библиотеку Python Seaborn вычислим корреляционную матрицу, которая будет указывать на связь между столбцами.

```
In [11]: sns.heatmap(df.corr(),annot=True)
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x79d0836f28>
```



Разделим базу данных на две функции, как столбцы данных для обучения и целевую функцию меток растений следующим образом.

#Столбцы = df[['temperature', 'humidity','ph', 'rainfall']]

features = df[['N', 'P','K','temperature','humidity', 'ph', 'rainfall']]

target = df['label'] labels = df['label']

Инициализация пустых списков для добавления всех названий моделей и соответствующих имён

acc = [] model = []

Разделение данных на обучающие и тестовые

from sklearn.model_selection import train_test_split

Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random_state =2)

Рассмотрим теперь технологии применения алгоритмов машинного обучения к обучению моделей и их анализу. Для анализа данных и построение моделей в системе sklearn есть множество реализованных алгоритмов машинного обучения. Ниже мы будем использовать эту систему для построения моделей и их оценки для анализа урожайности. Для удобства чтения, весь код написанный на Python будет иметь комментирии

1. Алгоритм дерево решений

   #Вызываем модуль алгоритма

   from sklearn.tree import DecisionTreeClassifier

   #Задаем параметры алгоритма дерево решений, критерий оценки ошибки #crossentropy, глубина дерева равно 5

   DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

#Задаем параметры алгоритма дерево решений,глубина дерева равно 5

#Обучим модель

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)

acc.append(x)

model.append('Decision Tree')

print("Точность алгоритма дерево решений: ", x*100,"%")

print(classification_report(Ytest,predicted_values))

```
Точность алгоритма дерево решений:  90.0 %
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      apricot       1.00      1.00      1.00        26
      barley       1.00      1.00      1.00        17
black currant       0.59      1.00      0.74        16
      cherry       1.00      1.00      1.00        29
      cotton       1.00      1.00      1.00        20
    cucumber       1.00      0.84      0.91        19
      grapes       1.00      1.00      1.00        18
  kidneybeans       0.00      0.00      0.00        14
      lentil       0.68      1.00      0.81        23
       maize       1.00      1.00      1.00        21

   mothbeans       0.00      0.00      0.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
        pear       1.00      1.00      1.00        22
  pigeonpeas       0.62      1.00      0.77        18
        plum       0.74      0.93      0.83        28
 pomegranate       1.00      1.00      1.00        17
      potato       1.00      0.62      0.77        16
      tomato       0.91      1.00      0.95        21
  watermelon       1.00      1.00      1.00        15
       wheat       1.00      1.00      1.00        21

    accuracy                           0.90       440
   macro avg       0.84      0.88      0.85       440
weighted avg       0.86      0.90      0.87       440
```

Теперь остановимся на технологии кроссвалидации. Проверим оценку точности на дерево решений глубины cv=5

```
from sklearn.model_selection import cross_val_score
```

Оценка перекрестной проверки кроссвалидации данных -дерево решений

```
score = cross_val_score(DecisionTree, features, target,cv=5)
score
array([0.93636364, 0.90909091, 0.91818182, 0.87045455, 0.93636364])
```

Сохранение построенной модели на основе дерево решений .

Сохраним модель с помощью pickle.Файл будет сохранен на диске в каталоге где находится сам проект.

```
import pickle
```

Выгрузите обученный классификатор с помощью Pickle

```
DT_pkl_filename = 'C:/Users/User/Desktop/A_notebooks_Urojai/models/DecisionTree.pkl'
```

Откройте файл, чтобы сохранить как файл pkl

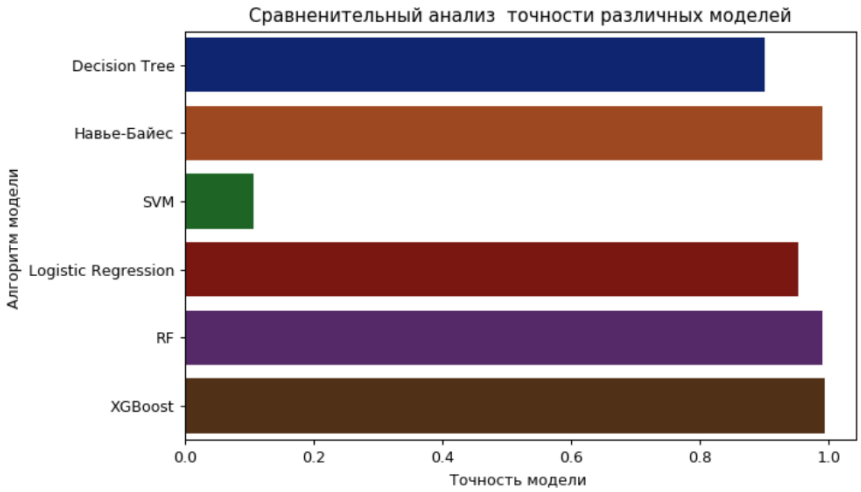
```
DT_Model_pkl = open(DT_pkl_filename, 'wb')
pickle.dump(DecisionTree, DT_Model_pkl)
```

Закройте экземпляры pickle

```
DT_Model_pkl.close()
```

В данной работе, мы построили модели урожайности для различных сельскохозяйственных культур в зависимости от различных признаков с применением алгоритмов машинного обучения дерево решений, Навье-Байеса и Гаусса, SVM, логистической регрессии, случайный лес и градиентный бустинг Среди них алгоритм Навье-Байеса и Гаусса

применяется не во всех случаях. Абсолютным лидером при построении моделей являются, алгоритмы случайный лес и градиентный бустинг. Данные алгоритмы для многих задач являются абсолютными победителями и является наиболее подходящими алгоритмами для построения сложных и точных моделей.



Вот результаты обучения моделей на основе проведенных численных расчетов. Приведем точности построенных моделей.

Ввод [39]:
```
accuracy_models = dict(zip(model, acc))
for k, v in accuracy_models.items():
    print (k, '-->', v)
```

```
Decision Tree --> 0.9
Навье-Байес --> 0.990909090909091
SVM --> 0.10681818181818181
Logistic Regression --> 0.9522727272727273
RF --> 0.990909090909091
XGBoost --> 0.9931818181818182
```

Теперь на алгоритме случайный лес делаем прогноз. Вот результаты.

```
data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
prediction = RF.predict(data)
print(prediction)
```

```
['pear']
```

```
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
prediction = RF.predict(data)
print(prediction)
```

```
['plum']
```

Результаты показывают, что прогноз выполняется почти на 100%.

Таким образом мы изучили основные влияющие факторы на урожайность сельскохозяйственных культур. Построили модели урожайности для различных сельскохозяйственных культур в зависимости от различных признаков с применением алгоритмов машинного обучения. Среди построенных моделей определили, что алгоритм случайный лес является наиболее подходящей для нашей цели.

Заключение

В данной работе изучается выявления основных признаков влияющих на урожайность сельскохозяйственных культур. С применением данных для определенного региона построены модели и прогнозы, определяющие достоверность построенных моделей на тестовых данных. Модели обучаются на современных алгоритмах машинного обучения.

ЛИТЕРАТУРА

1. Орельен Жерон − Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow, 2018 г.
2. Гудфеллоу Я., Бенджио И., Курвилль А. − Глубокое обучение, 2017 г.
3. Дж. Вандер Плас – Python для сложных задач. Наука о данных и машинное обучение,2020.
4. Ричард Саттон, Эндрю Барто − Обучение с подкреплением, 2017 г.
5. Андрей Бурков − The Hundred-Page Machine Learning Book, 2019 г.
6. Максим Лапань − Deep Reinforcement Learning Hands-On, 2018 г.