



**САБИТОВ Б.Р., ОРОЗОБЕКОВА А.К., ЖАДИЛОВ Б.М., СЕЙТБЕКОВ А.,  
ШАМЫРОВА Д.Р., ШЕРИМБЕКОВА Э.Б.**

<sup>1</sup> Кыргызский национальный университет им. Ж.Баласагына, Бишкек,  
Кыргызская Республика

<sup>2</sup> КГУСТА им. Н.Исанова, Бишкек, Кыргызская Республика  
**SABITOV B.R., OROZOBEKOVA A.K., ZHADILOV B.M., A. SEITBEKOV A.,  
SHAMYROVA D.R., SHERIMBEKOVA E.B.**

<sup>1</sup> Kyrgyz National University named after J. Balasagyn, Bishkek, Kyrgyz Republic  
<sup>2</sup> KSUCTA n.a. N. Isanov, Bishkek, Kyrgyz Republic

### **ПРИМЕНЕНИЕ СИСТЕМЫ FBPROPHET НА БАЗЕ ТЕХНОЛОГИЙ МАШИННОГО ОБУЧЕНИЯ ПРИ ПРОГНОЗИРОВАНИИ ЗАДАЧ АПК**

### **APPLICATION OF THE FBPROPHET SYSTEM BASED ON TECHNOLOGIES OF MACHINE LEARNING AND TIME SERIES FORECASTING AIC TASKS**

*Бул макалада агроөнөр жай комплексиндеги (АӨК) маселелерди болжолдоодо машина үйрөнүү технологияларын жана убакыт катарлардын негизде каралат. Түшүмдүүлүк көйгөйлөрүн болжолдоо процесси FbProphet заманбап системасынын жардамы менен изилденген жана Ысык-Көл облусунун айрым региондору үчүн айыл чарба өсүмдүктөрүнүн түшүмдүүлүгү боюнча реалдуу маалыматтар боюнча божомолдун жыйынтыктары алынган.*

**Өзөк сөздөр.** *Агроөнөр жай комплекси, машина үйрөнүү, убакыт катарлары, болжолдоо системасы, айыл чарба өсүмдүктөрү.*

*В данной статье исследуется прогнозирования задач агропромышленного комплекса (АПК) на базе технологий машинного обучения и временных рядов. Изучен процесс прогнозирования задач урожайности с применением современной системы FbProphet и получены результаты прогноза на реальных данных по урожайности сельскохозяйственных культур для отдельных регионов Исык-Кульской области.*

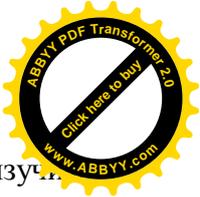
**Ключевые слова.** *АПК, машинное обучение, временные ряды, система прогнозирования, сельскохозяйственные культуры*

*This article examines the forecasting of tasks in the agro-industrial complex (AIC) based on machine learning technologies and time series. The process of forecasting yield problems using the modern FbProphet system is studied and the results of the forecast based on real data on the yield of agricultural crops for individual regions of the Issyk-Kul region are obtained.*

**Key words.** *Agro-industrial complex, machine learning, time series, forecasting system, crops.*

В данной работе изучим задачу прогнозирования временных рядов. Прогнозирование временных рядов, является одним из сложнейших задач для ее использования в прикладных целях. Она связано с изучением реальных нелинейных процессов, происходящих в природе, например, выявление природных аномалий, предсказание селевых и оползневых процессов, задачи фондового рынка.

Facebook создал мощнейшую интеллектуальную систему FbProphet (Пророк) для задач прогнозирования временных рядов, которые описывают сложные нелинейные процессы многих прикладных задач. Данная система основано на библиотеках и



технологиях Python. Ее можно реализовать и на языке R. В данной работе мы изучим данную систему для прогнозирования и построения нелинейных моделей с применением технологий машинного обучения, применительно к задачам сельского хозяйства. Изучим, также вопросы использования библиотек FbProphet для прогнозирования временных рядов.

В отличие от других библиотек в системе FbProphet значительно упрощен процесс прогнозирования временных рядов, и она интуитивно понятна для ее использования для многих задач прогнозирования. В целях программной реализации нам необходимо сначала установить программный пакет FbProphet с ее необходимыми библиотеками и нужно создать также, объект класса Prophet. Необходимо загрузить изучаемую базу данных в виде .csv файла. При этом необходимо, также настроить множество параметров используемые в Prophet, для ее полноценной работы. Например, нужно указать диапазон интервала неопределенности с помощью параметра `interval_width`.

Установим интервал неопределенности в 95%, так как по умолчанию для Prophet 80%.

```
my_model = Prophet(interval_width=0.95)
```

Теперь библиотека Prophet может работать с временными рядами. Затем нужно вызвать метод `fit` созданного объекта т.е . обучит нашу будущую модель с помощью метода `fit`

```
my_model.fit(df)
```

По требованию данной системы библиотека Prophet, входная информация в виде DataFrame должна содержать столбец с информацией о времени, и столбец, содержащий метрику, которую нужно спрогнозировать. Важно отметить, что столбец времени должен содержать данные типа `datetime`. Чтобы получить прогноз временного ряда, предоставим библиотеке Prophet новый DataFrame, содержащий столбец `ds`. Этот DataFrame не нужно создавать вручную, поскольку Prophet предоставит нам функцию.

```
make_future_dataframehelper  
future_dates = my_model.make_future_dataframe(periods=36, freq='MS')  
future_dates.tail()
```

Вот результат:

	ds
175	2021-08-01
176	2021-09-01
177	2021-10-01
178	2021-11-01
179	2021-12-01

Обычно система Prophet генерирует 36 меток времени. При использовании Prophet, также важно указать частоту временных рядов, `freq`. Для работы с данными вводится также, с данные за месяц, в коде мы ее указываем как `MS` (это начало месяца). Функция `make_future_dataframe` при этом будет генерировать 36 ежемесячных временных отметок.

Таким образом мы в данном случае получим прогнозирования на три года вперед. Данный фрейм-класс, имеет при этом метод прогноза `predict`. С помощью данного метода мы можем прогнозировать временной ряд по обученной модели. Вот результат прогноза временного ряда.

```
forecast = my_model.predict(future_dates)  
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```



	ds	yhat	yhat_lower	yhat_upper
175	2021-08-01	108.475630	50.110126	169.558586
176	2021-09-01	93.593173	34.345564	147.460111
177	2021-10-01	126.492673	68.924136	186.411487
178	2021-11-01	101.828911	41.348613	159.434157
179	2021-12-01	112.233843	56.287321	164.467313

Система Prophet при этом возвращает данные с большим количеством столбцов. Перечислим эти столбцы:

- ds: временная метка прогнозируемого значения.
- yhat: прогнозируемый признак и обозначается как yhat.
- yhat\_lower: нижняя граница прогнозов.
- yhat\_upper: верхняя граница прогнозов.

Ожидается отклонение в значениях от представленного выше результата, поскольку Prophet использует методы MS (начало месяца) для получения своих прогнозов. MS – это стохастический процесс. При этом мы в результате получим данные с погрешностью. Система Prophet имеет хорошую функцию для визуализации прогнозируемых данных. Также Prophet показывает значения временных рядов:

- черные точки, прогнозируемые значения;
- синяя линия и интервалы неопределенности прогнозов;
- синие заштрихованные области.

Для нашего случая пишем следующий код.

```
my_model.plot(forecast, uncertainty=True)
```

В результате мы получим прогноз урожайности картофеля за 2 года вперед.

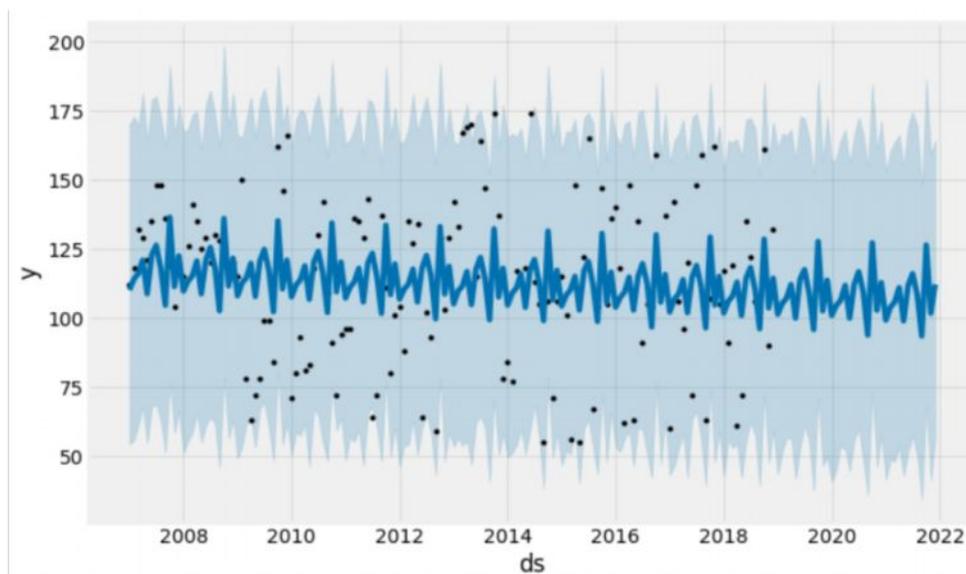


Рис. 1. Результат прогноза урожайности картофеля с использованием FbProphet

Одним из преимуществ функции Prophet – это способность возвращать компоненты прогнозирования. Это помогает выявлять элементы прогнозирования с применением Prophet. Таковыми являются ежедневные, недельные и годовые модели временных рядов. Представим их визуализацию с помощью метода plot.

```
my_model.plot_components(forecast)
```

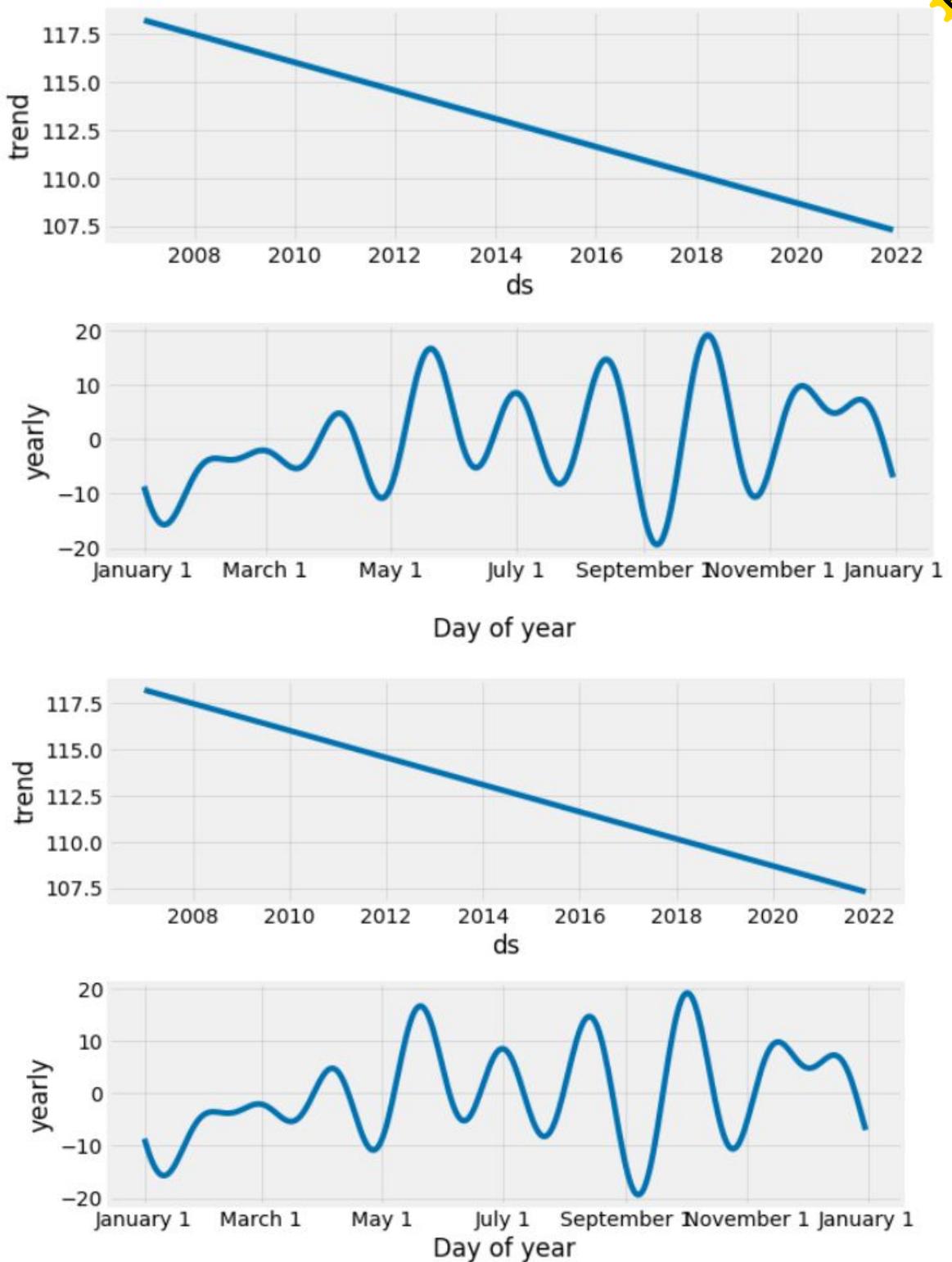


Рис. 2. Графическое представление компонентов прогноза урожайности картофеля с использованием FbProphet

Полученные графики предоставляет много информации о прогнозировании. В графиках указано, что ежемесячный урожайность линейно уменьшается с течением времени до следующего урожая. Также имеется информация что, к концу недели количество урожая увеличивается; на графиках показано, что наибольшее количество перемещений приходится на период сбора урожая.

Python API



Используя API sklearn FbProphet можно создать модели. Для этого, мы создаем экземпляр класса Prophet, а затем вызываем его методы fit и Foret. Входные данные для Prophet всегда представляют собой фрейм данных с двумя столбцами: ds и y. Столбец ds (datetime) должен иметь формат даты библиотеки Pandas, в идеале YYYY-MM-DD для даты или YYYY-MM-DD HH: MM: SS для отметки времени. Столбец y должен быть числовым и представлять измерение, которое мы хотим прогнозировать. В качестве примера, рассмотрим временные ряды с большими данными, который является хорошим примером, т.к. иллюстрирует некоторые особенности Пророка, таких как множественная сезонность, изменяющиеся темпы роста и способность моделировать особые дни. Сначала импортируем данные:

```
import pandas as pd
from fbprophet import Prophet
```

Обращаемся к базе данных об урожайности картофеля в тоннах с гектара временного ряда

```
df = pd.read_csv('Urojaj_kart.csv')
df.head(20)
```

Библиотека Pandas всегда отображает данные с 6-ю знаками после запятой. Всего объектов в базе данных

```
df.shape
(2905, 2)
```

Вызываем необходимые библиотеки.

```
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

Строим с этими данными график изменения данных. Вот код визуализации данных.

```
ax = df.set_index('ds').plot(figsize=(12, 8))
ax.set_ylabel('Урожайность картофеля тонна с га')
ax.set_xlabel('Data')
plt.show()
```

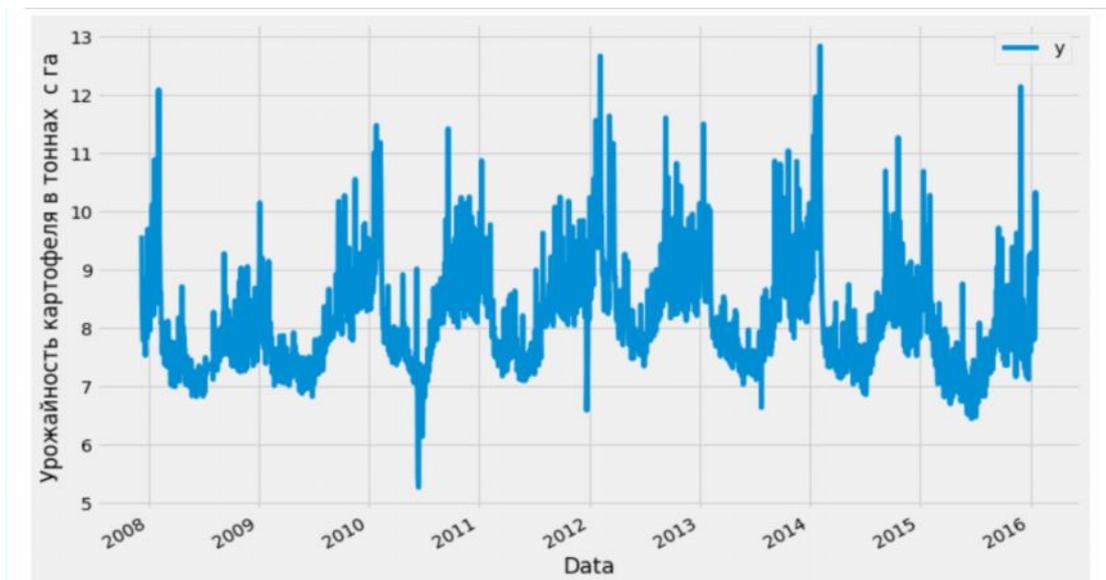


Рис.3. Представление изменения урожайности картофеля по годам с использованием FbProphet

Как и система sklearn, которая является интеллектуальной системой Python, библиотека Prophet имеет похожий интерфейс, сначала она, создает модель, затем вызывает метод обучения fit, и потом только получаем прогнозируемое значение. При этом на вход методу predict.fit библиотека принимает dataframe с двумя колонками: ds — время, данное поле должно быть строго типа date или datetime,



$y$  — это прогнозируемый признак т.е. в нашем случае числовой показатель, который мы хотим прогнозировать. Теперь мы подгоняем модель, создавая экземпляр нового объекта Prophet. Любые настройки процедуры прогнозирования передаются в конструктор. Затем вызывается метод `fit` и передается исторический фрейм данных.

```
m = Prophet () m.fit(df)
```

Прогнозы затем делаются на кадре данных со столбцом `ds`, содержащим даты, на которые должен быть сделан прогноз. Мы можем также, получить подходящий фрейм данных, который простирается в будущем на указанное количество дней, используя вспомогательный метод `Prophet.make_future_dataframe`. По умолчанию он также будет включать в себя даты из истории, поэтому мы также увидим, как модель подходит.

```
future = m.make_future_dataframe(periods=1395)
future.tail(20)
```

	ds
4280	2019-10-27
4281	2019-10-28
4282	2019-10-29
4283	2019-10-30
4284	2019-10-31
4285	2019-11-01
4286	2019-11-02
4287	2019-11-03
4288	2019-11-04
4289	2019-11-05
4290	2019-11-06
4291	2019-11-07
4292	2019-11-08
4293	2019-11-09
4294	2019-11-10

Метод `predict` — прогнозирование присваивает каждой строке в «будущем» прогнозируемое значение, которое он называет «`yhat`». Если мы укажем исторические даты, это даст примерную подгонку. Объект «прогноз» здесь представляет собой новый фрейм данных, который включает столбец «`yhat`» с прогнозом, а также столбцы для компонентов и интервалов неопределенности.



```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(10)
```

	ds	yhat	yhat_lower	yhat_upper
4290	2019-11-06	6.378644	3.329593	9.309761
4291	2019-11-07	6.361834	3.330033	9.138821
4292	2019-11-08	6.356316	3.206198	9.132136
4293	2019-11-09	6.109160	3.030489	8.897744
4294	2019-11-10	6.467476	3.388022	9.206409
4295	2019-11-11	6.773160	3.709161	9.561158
4296	2019-11-12	6.545529	3.386532	9.236935
4297	2019-11-13	6.367448	3.335503	9.215253
4298	2019-11-14	6.373026	3.308339	9.207129
4299	2019-11-15	6.389590	3.380963	9.114782

Теперь мы построим прогноз, вызвав метод Prophet.plot и передав свой фрейм данных прогноза.

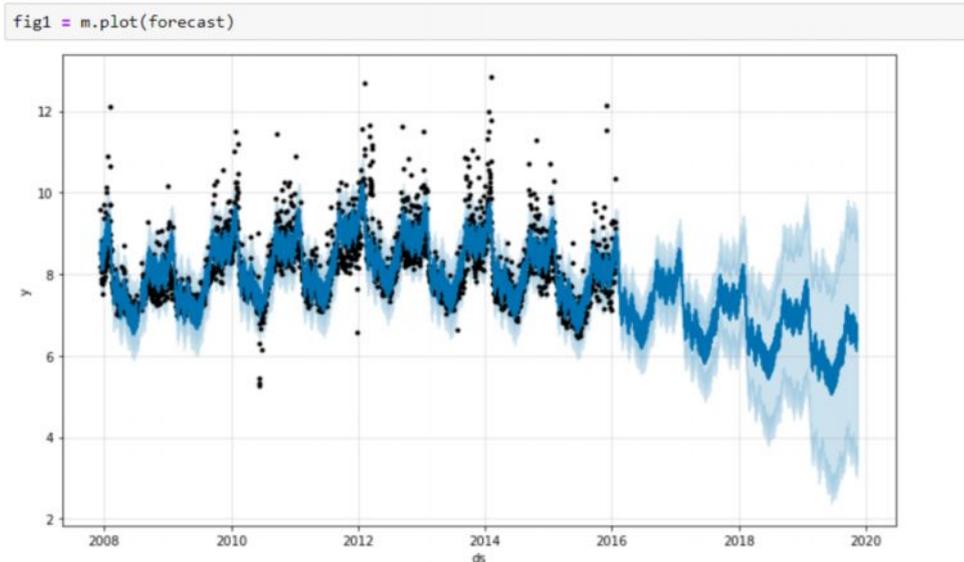


Рис. 4. Прогнозирование урожайности картофеля по годам с использованием FbProphet

Выше на графике интеллектуальная система Prophet предоставляет нам, удобную форму представления данных прогноза для визуализации и показывает значения временных рядов:

- черные точки, прогнозируемые значения;
- синяя линия и интервалы неопределенности прогнозов;
- синие заштрихованные области.

Далее система вычисляет компоненты прогноза. Для этой цели используется метод Prophet.plot\_components. Ниже представлены тренды годовые, недельные и дневные сезонности прогнозируемого временного ряда.

```
fig2 = m.plot_components(forecast)
```

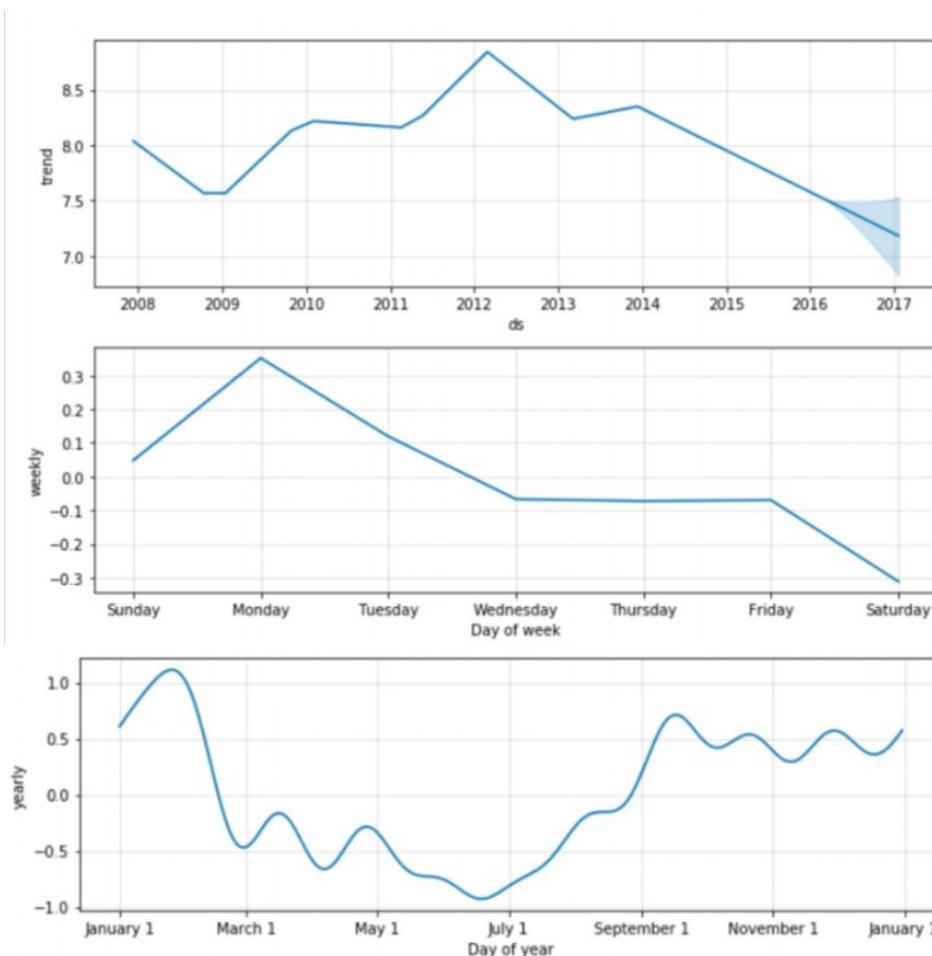


Рис. 5. Тренды изменения данных прогнозирования по годам, неделям и дням построенная системой FbProphet

**Заключение.** В данной работе изучены различные методы прогнозирования временных рядов с применением технологий FbProphet. Изучена новейшая технология работы с временными рядами- интеллектуальная система FbProphet. Ознакомились с технологией современного прогнозирования, не прибегая к большим трудностям, как другие статистические методы прогнозирования для прикладных задач. Система прогнозирования FbProphet, считается современной и универсальной технологией сегодняшнего дня, наряду с XGBoost –градиентный бустинг машинного обучения и удобный инструмент для управления данными временных рядов, а также ее визуализации с помощью библиотек Python.

### Список литературы

1. Орельен Ж. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow [Текст] / ЖД.Орельен // Переводчик Ю.Н. Артеменко. – М.: Диалектика, 2020. 1040 с.
2. Ричард Саттон Обучение с подкреплением [Текст] / Ричард Саттон, Эндрю Барто // Переводчик А.А.Слинкин. – М.: ДМК-Пресс, 2020.- 552 с.
3. Официальный сайт scikit-learn: <https://scikit-learn.org>
4. Официальный сайт FbProphet: <https://prophet.com>