

NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science and Business Analytics"

Report
On Academic Internship Results
in Higher School of Economics
Theme: "Prediction of Average Salaries by Region in 2025–2026"

Fulfilled by the student of the group 234

Doronicheva Polina Andreevna



Internship Supervisor:

Popov Victor Yurievich
Professor, Head of Laboratory
Faculty of Computer Science, HSE University



Contents

Introduction	3
Objectives	3
Research Plan	3
Calendar Schedule	4
1 Research Methodology	5
1.1 Data Collection	5
1.2 Data Pre-Processing	5
1.3 LSTM Model Construction	6
1.4 Model Training and Evaluation	7
1.5 Forecasting and Visualization	8
2 Conclusion	13
2.1 Analysis of changes in salaries	13
2.2 Limitations of the Study	13
2.3 Future Work	14
References	15

Introduction

This project aims to forecast average wages in Russian regions for 2025-2026 using machine learning. Historical data from Rosstat for the period from 2013 to 2024 were used for data analysis. Forecasting was performed using the LSTM neural network adapted for time series analysis.






Objectives

The aim of the paper is to create a model suitable for forecasting wages in Russian regions in 2025-2026 based on old data, as well as subsequent evaluation of the forecast results, including analyzing changes in average wages in different regions, identifying trends and predicting possible changes in the future.

Research Plan

1. Collecting data: Downloading official monthly wage statistics by region from the Rosstat website.
2. Cleaning and preparing data: Removing unnecessary rows, merging duplicate regions, and calculating annual averages.
3. Encoding and scaling: Applying one-hot encoding to categorical variables and normalizing the data with `MinMaxScaler`.
4. Splitting the dataset: Dividing the data into training and test sets based on historical time periods.
5. Building the model: Designing an LSTM-based neural network using Keras.
6. Training and evaluating the model: Fitting the model to training data and assessing performance using MAE and MSE.
7. Forecasting: Generating wage predictions for 2025 and 2026 and calculating a forecast interval.
8. Visualizing results: Creating plots to show trends, regional differences, and the distribution of forecasted changes.

Calendar Schedule

№	Calendar period	Plan of work	Supervisor's mark on the point fulfilment (signature)
1	01.07.2025	1. Instructing on the requirements of labor protection, safety, fire safety and internal labor regulations	
2	01.07.2025	2. Organizational (induction) meeting	
3	02–10.07.2025	3. Fulfilment of Individual Assignment	
4	11.07.2025	4. Consultation	
5	12–14.07.2025	5. Preparation and submission of the Report	

1 Research Methodology

1.1 Data Collection

For this paper data was obtained from the official statistical resource Rosstat [2], which is the main source of official statistical information in Russia. In particular, the dataset “Average monthly nominal gross nominal wages of employees in the full range of organizations by constituent entities of the Russian Federation since 2013 (by month), rubles” was used. These data is presented in Excel format and require additional cleaning and/or merging of some columns. In my GitHub, this table is named “wages.xlsx”.

1.2 Data Pre-Processing

First, several unnecessary rows were removed from the table, and the regions "Kuzbass" and "Kemerovo Oblast" were merged under a single name to ensure consistency. (fig. 1.1)

```
def get_salary(df, suffix_data):
    result = defaultdict(list)

    for _, row in df.iterrows():
        region = row.iloc[0]
        if not isinstance(region, str):
            continue
        region = region.strip()
        if "Кузбасс" in region:
            region = "Кемеровская область"
        if "в том числе" in region or "федеральный округ" in region or "1)" in region or "2)" in region:
            continue
        for col in df.columns[1:]:
            match = re.match(r'^(' + '|'.join(months) + r')(\.\d*)?$', col)
            if match:
                _, suffix = match.groups()
                suffix = suffix or ''
                year = suffix_data.get(suffix)
                if year:
                    value = row[col]
                    if pd.notna(value):
                        result[(region, year)].append(value)
```

Figure 1.1: Extracting and cleaning salary data from the source file

Then, since the data set provided monthly average salaries, the average annual salary was calculated for each region. The resulting data frame consisted of three columns: "Region", "Year", and "Mean Wage". (fig. 1.2)

The data were then transformed and split into training and test samples. First, categorical variables (Region and Year) were converted into numerical format using one-hot encoding via the `get_dummies()` function.

After encoding, the dataset was divided into training and test samples (fig. 1.3). The training set included

data from 2017 to 2024, and the test set contained data from 2013 to 2016. This approach has helped to reduce mean absolute error (MAE) and mean squared error (MSE), which will be discussed later.

	Region	Year	Mean_Wage
0	Алтайский край	2013	18116.933333
1	Алтайский край	2014	19456.933333
2	Алтайский край	2015	19959.758333
3	Алтайский край	2016	21039.441667
4	Алтайский край	2017	22733.493642
...
1035	г.Севастополь	2020	35660.991667
1036	г.Севастополь	2021	39127.000000
1037	г.Севастополь	2022	43732.833333
1038	г.Севастополь	2023	52720.900000
1039	г.Севастополь	2024	60490.833333

Figure 1.2: Initial structure of the dataset

```
In [5]:
x = pd.get_dummies(df_salary[['Region', 'Year']])
y = df_salary['Mean_Wage']

In [6]:
flag = df_salary['Year'] >= 2017

xTrain = x[flag]
xTest = x[~flag]

yTrain = y[flag]
yTest = y[~flag]
```

Figure 1.3: Splitting data into training and test samples

1.3 LSTM Model Construction

Before training the model, all input and output data were normalized using the MinMaxScaler function. This transformation scaled the features into the range [0, 1] to improve the accuracy of the neural network. Two separate scalers were used: one for the features (x) and one for the target variable (y). (fig. 1.4)

```
xScaler = MinMaxScaler()
xTrainScaler = xScaler.fit_transform(xTrain)
xTestScaler = xScaler.transform(xTest)

yScaler = MinMaxScaler()
yTrainScaler = yScaler.fit_transform(yTrain.values.reshape(-1, 1))
yTestScaler = yScaler.transform(yTest.values.reshape(-1, 1))

xTrainLSTM = xTrainScaler.reshape(xTrainScaler.shape[0], 1, xTrainScaler.shape[1])
xTestLSTM = xTestScaler.reshape(xTestScaler.shape[0], 1, xTestScaler.shape[1])
```

Figure 1.4: Scaling and reshaping the data

After scaling, the input arrays were reshaped to match the expected format of LSTM network. Specifically, each feature set was reshaped into a three-dimensional array with the shape (samples, time steps, features), where the number of time steps was set to 1 to reflect the structure of the historical data used for prediction. (fig. 1.4)

The LSTM model was built using the Keras Sequential API. It includes an LSTM layer with 64 units and ReLU activation; a Dropout layer (rate 0.2) to reduce overfitting; a Dense layer with 32 units and ReLU activation; a second Dropout layer; and a single-unit Dense output layer. The model was compiled with the Adam optimizer, using MSE as the loss function and MAE as a performance metric. (fig. 1.5)

```
model = Sequential()
model.add(Input(shape=(xTrainLSTM.shape[1], xTrainLSTM.shape[2])))
model.add(LSTM(64, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1))

model.compile(optimizer='adam', loss='mse', metrics=['mae'])
```

Figure 1.5: LSTM model architecture

1.4 Model Training and Evaluation

The model was trained using the `fit()` function with 9 epochs and a batch size of 32. A test set was used as validation data to monitor model performance. Also MAE and MSE were tracking during training. (fig. 1.6)

```
history = model.fit(xTrainLSTM, yTrainScaler,
                    epochs=9,
                    batch_size=32,
                    validation_data=(xTestLSTM, yTestScaler))
```

Figure 1.6: Model Training

MAE measures the average absolute difference between predicted values and actual values. In my case, it shows how much on average the salary predicted by the model differs from the real one (in rub).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

MSE is the mean value of the square of the error. In my case, it shows how much the wage predicted by the model differs from the real wage on average (in rub), with an emphasis on large errors.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Throughout the training process, both metrics decreased on the training and validation sets (fig. 1.7). By the final epoch, the MAE had reached a low and stable value, indicating that the model was able

to generalize well and make accurate predictions across different regions and years. The consistency between training and validation MAE also suggests that overfitting was successfully mitigated.

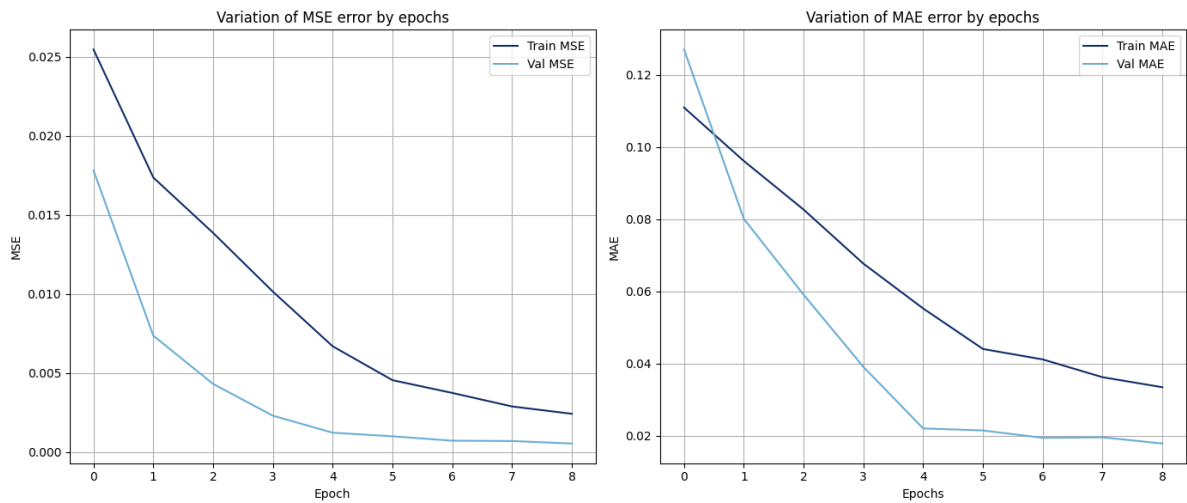


Figure 1.7: MSE and MAE tracking

1.5 Forecasting and Visualization

To forecast average wages for the years 2025 and 2026, a new dataset was constructed. It had all unique regions from the original data and two future years: 2025 and 2026. The categorical features (Region, Year) were then one-hot encoded using the same structure as in the training data to ensure consistency. (fig. 1.8)

```
xFuture = pd.get_dummies(df_future_salaries)
xFuture = xFuture.reindex(columns=x.columns, fill_value=0)
xFutureScaler = xScaler.transform(xFuture)
xFutureLSTM = xFutureScaler.reshape(xFutureScaler.shape[0], 1, xFutureScaler.shape[1])
```

Figure 1.8: Encoding

Once prepared, the input data were passed into the trained model to generate predictions. The output was inverse-transformed back into the original salary scale. To estimate uncertainty in predictions, a margin based on the model's MAE (MAE was calculated on test sample, as I did not have right values for the future years) was added and subtracted to calculate minimum and maximum forecast intervals. (fig. 1.9)

```
yFutureScaler = model.predict(xFutureLSTM)
yFuture = yScaler.inverse_transform(yFutureScaler)
df_future_salaries['Forecast_Wage'] = yFuture

df_future_salaries['Min_Forecast_Wage'] = df_future_salaries['Forecast_Wage'] - mae
df_future_salaries['Max_Forecast_Wage'] = df_future_salaries['Forecast_Wage'] + mae
```

Figure 1.9: Forecast

To visualize the forecasted data I used interactive line plot. This plot allows the user to select a region and year (2025 or 2026) and see the average wage trend from 2013 to the selected year, including both historical data and predicted values.

For illustration purposes, the graph below shows the salary trend for the Moscow region. It demonstrates that in 2025 year the mean wage for Moscow will decrease to about 150000 rubles, but in 2026 a mean wage will be about 160000 rubles. (This widget is not supported by GitHub, you need to download the file to see it and/or to change the region/year).

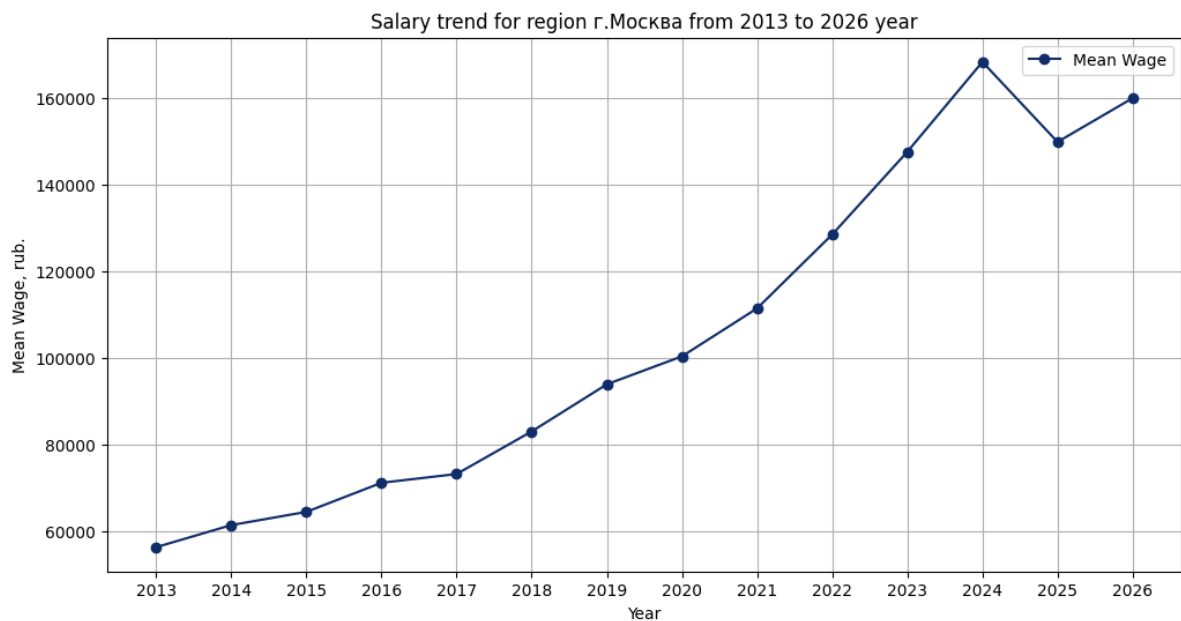


Figure 1.10: Trends for Moscow 2013-2026

Also, you can download the summary table with all the data on my GitHub. Here are some of the obtained data:

	Region	Year	Mean_Wage	Min_Forecast_Wage	Max_Forecast_Wage
0	Алтайский край	2013	18116.933333	NaN	NaN
1	Алтайский край	2014	19456.933333	NaN	NaN
2	Алтайский край	2015	19959.758333	NaN	NaN
3	Алтайский край	2016	21039.441667	NaN	NaN
4	Алтайский край	2017	22733.493642	NaN	NaN
...
1209	г.Севастополь	2022	43732.833333	NaN	NaN
1210	г.Севастополь	2023	52720.900000	NaN	NaN
1211	г.Севастополь	2024	60490.833333	NaN	NaN
1212	г.Севастополь	2025	70667.085938	67532.671875	73801.500000
1213	г.Севастополь	2026	79577.507812	76443.093750	82711.921875

Figure 1.11: Final table

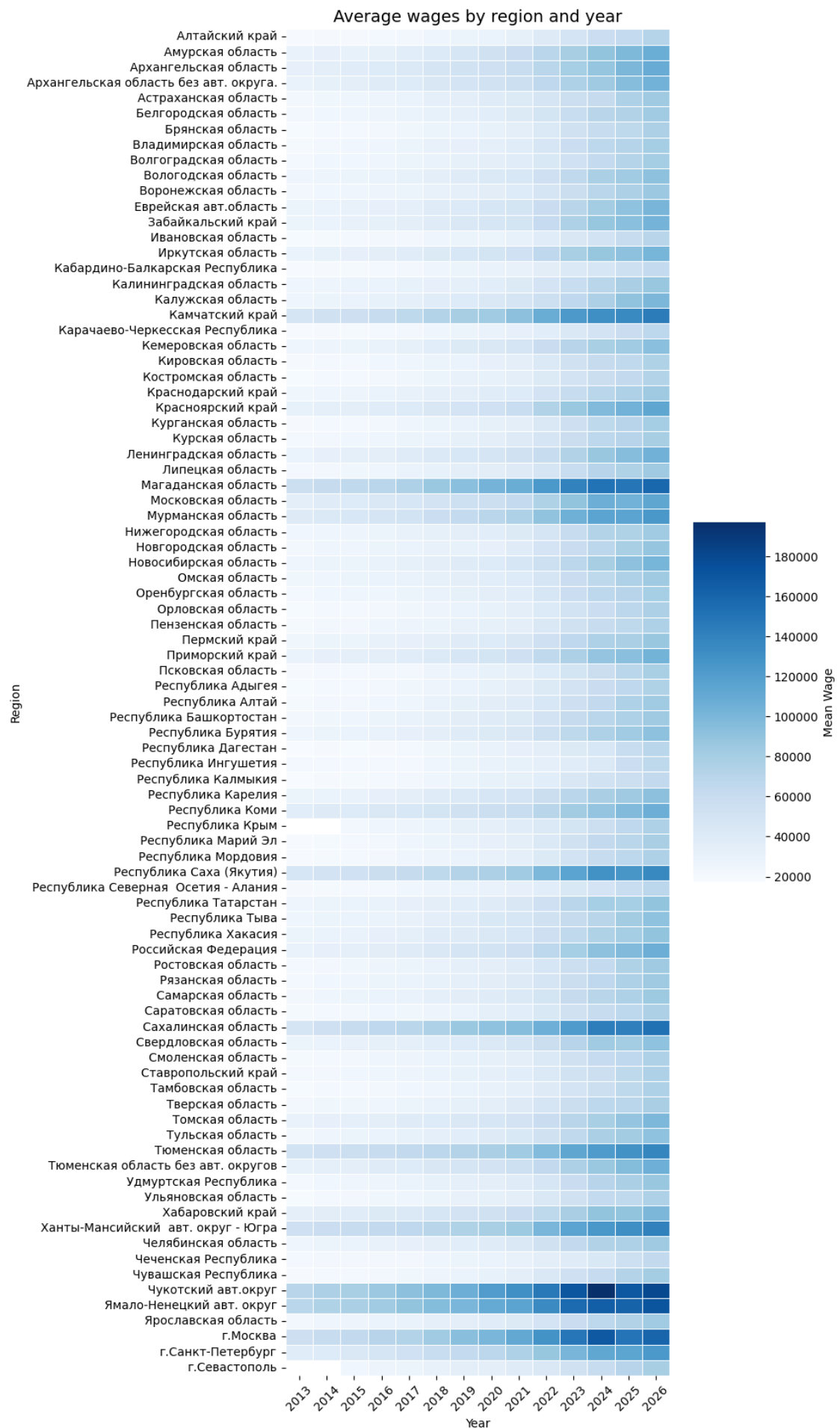


Figure 1.12: Heatmap of average wages by region and year

To show the average wage levels by region and year I used heatmap (fig. 1.12). Each cell represents the average wage in a specific region for a given year, with the color intensity reflecting the wage amount — the darker the color, the higher the salary.

This heatmap demonstrates the general growth of salaries over time. Also it emphasizes the differences in salaries by region. For example, darker areas in the lower part of the chart correspond to regions like Moscow, Yamalo-Nenets Autonomous Okrug, and Chukotka, where average wages are among the highest.

In addition to the heat map, a horizontal histogram chart showing the top 5 regions by salaries in 2025-2026 was created (fig. 1.13). The regions with the highest salaries include Chukotka Autonomous Okrug, Yamalo-Nenets Autonomous Okrug, Moscow, Magadan Oblast and Sakhalin Oblast.

The projected salary leaders for 2025 and 2026 look realistic and correspond to known economic patterns. Northern and resource-rich regions such as Chukotka Autonomous Okrug, Yamalo-Nenets Autonomous Okrug and Magadan Oblast show the highest projected salaries, which corresponds to traditionally high wages due to harsh climatic conditions and the presence of extractive industries. Moscow is also among the leading regions, reflecting its status as the country’s financial and administrative center. In general, the survey results are consistent with the actual distribution of salaries across Russia.

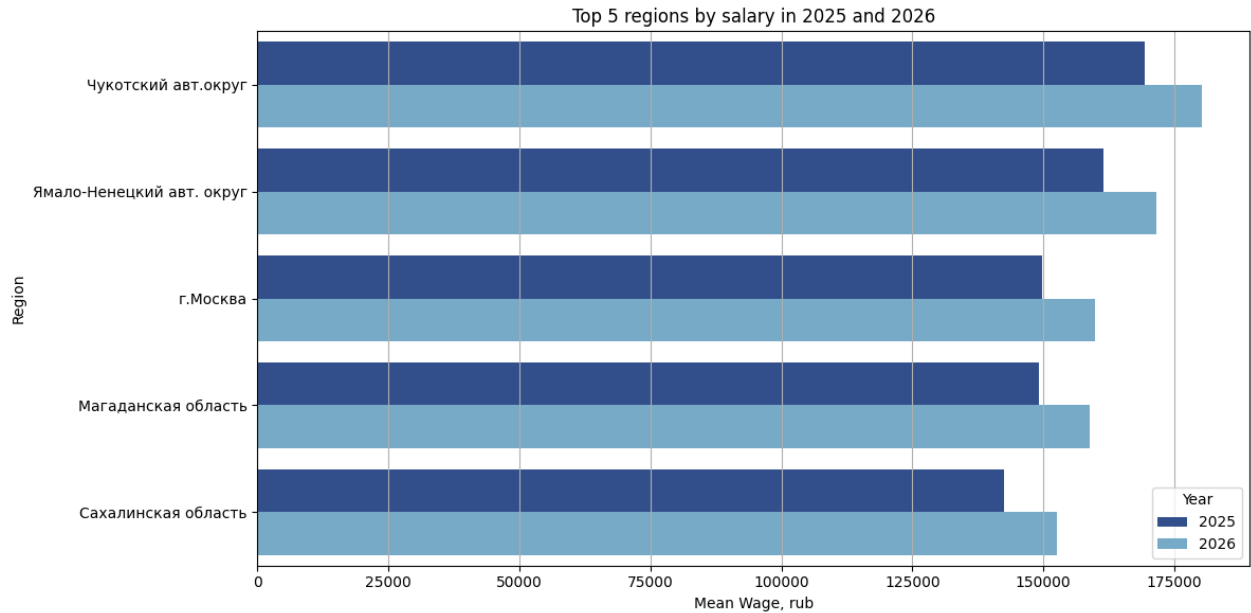


Figure 1.13: Top regions 2025-2026

To track the general trend in Russia, boxplot was created for each year from 2013 to 2026 (fig. 1.14). As can be seen from the figure, the median wage is growing from year to year, and the spread between them is noticeably increasing. This indicates not only the general growth of salaries, but also the growing inequality of average incomes between regions.

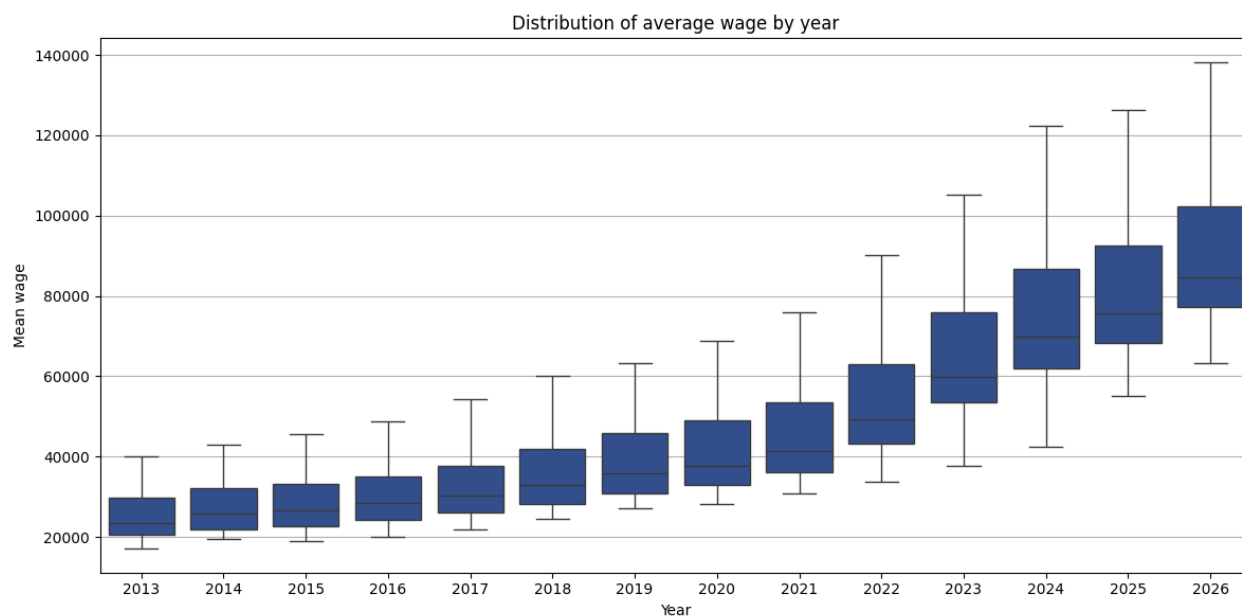


Figure 1.14: Boxplot of average wages 2013-2026

2 Conclusion

2.1 Analysis of changes in salaries

To better understand how wages are expected to change across regions, the percentage difference in average salary was calculated year-over-year. The histograms below show how many regions fall into each percentage change interval for 2025 and 2026.

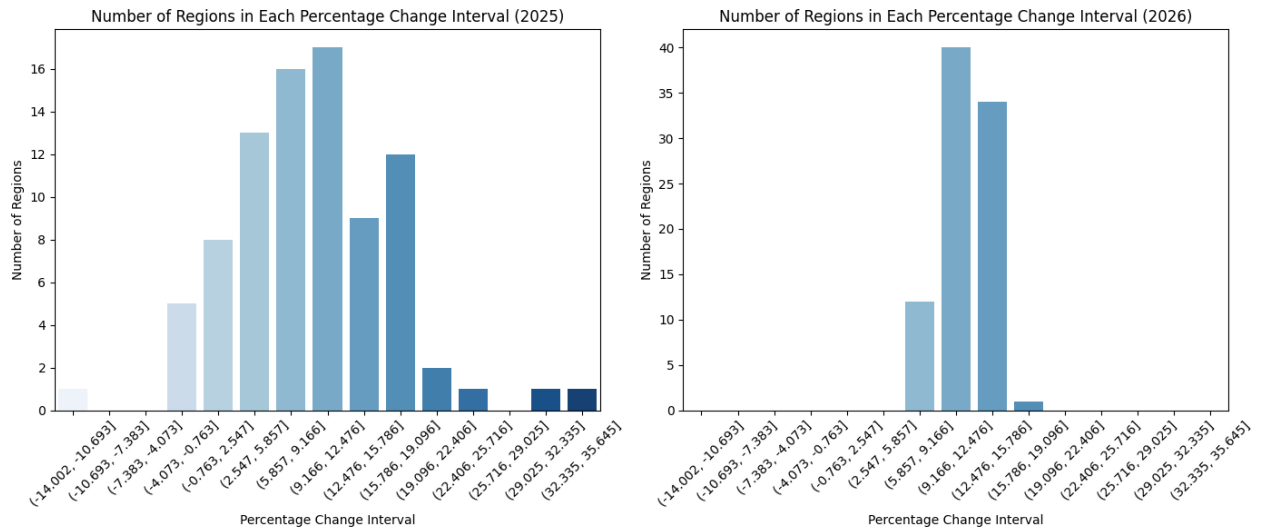


Figure 2.1: Distribution of percentage change in average salary

Between 2024 and 2025, a significant number of regions demonstrated wage growth in the range of approximately 0% to 15%, although several regions experienced a decline of around 4% to 7%. In the following period, from 2025 to 2026, the distribution of wage changes was more stable, with the majority of regions exhibiting growth between 5% and 12% and there was no negative growth.

2.2 Limitations of the Study

- The dataset included only average salaries by region and month. Other important factors such as inflation, cost of living, unemployment, or economic activity were not considered.
- All salary values are nominal. The model does not take inflation into account, so the predicted growth may partly reflect price increases rather than real income growth.
- The data were aggregated by year, based on monthly averages. Because of this, seasonal changes and short-term trends were not captured.
- Regional differences were represented in a simplified way using one-hot encoding. This does not reflect deeper economic or geographic relationships between regions.

2.3 Future Work

- Include additional features such as inflation rates, unemployment, regional GDP, or industry structure to make the forecasts more accurate.
- Adjust all salaries for inflation to provide forecasts in real terms rather than nominal values.
- Use more advanced encoding methods to better represent regional similarities and dynamics.
- Increase the size and variety of the dataset by incorporating longer time periods or additional data sources.
- Explore alternative forecasting models, attention-based networks, or hybrid approaches that combine machine learning with economic modeling.

References

- [1] My GitHub. Prediction of Average Salaries by Region in 2025–2026. 2025. URL: <https://github.com/meomato/Prediction-of-Average-Salaries-by-Region-in-2025-2026>.
- [2] Rosstat. Official Website of Rosstat Statistics. 2025. URL: https://rosstat.gov.ru/labor_market_employment_salaries.
- [3] Popov Victor Yurievich. Intro to Python and Google Colab. 2025. URL: https://drive.google.com/drive/folders/1DGCop-HP0SmgwE-GsHyih58ALWilQM_K?usp=share_link.
- [4] Popov Victor Yurievich. Neural Network: Materials and Sample Topics. 2025. URL: https://drive.google.com/drive/folders/1I2Jln1e4tPh7Ixp_Vex5P120a7lYpXxT?usp=sharing.
- [5] Popov Victor Yurievich. Neural Networks for Regression and Time Series. 2025. URL: https://drive.google.com/drive/folders/1mst8gtSlb278mtYCEIe-bL_Ovtdnkx1s?usp=share_link.
- [6] Popov Victor Yurievich. Text Processing with Neural Networks: Materials and Ideas. 2025. URL: https://drive.google.com/drive/folders/196ddJMap0J6trWvCFDafpNFOHcPdCyZl?usp=share_link.