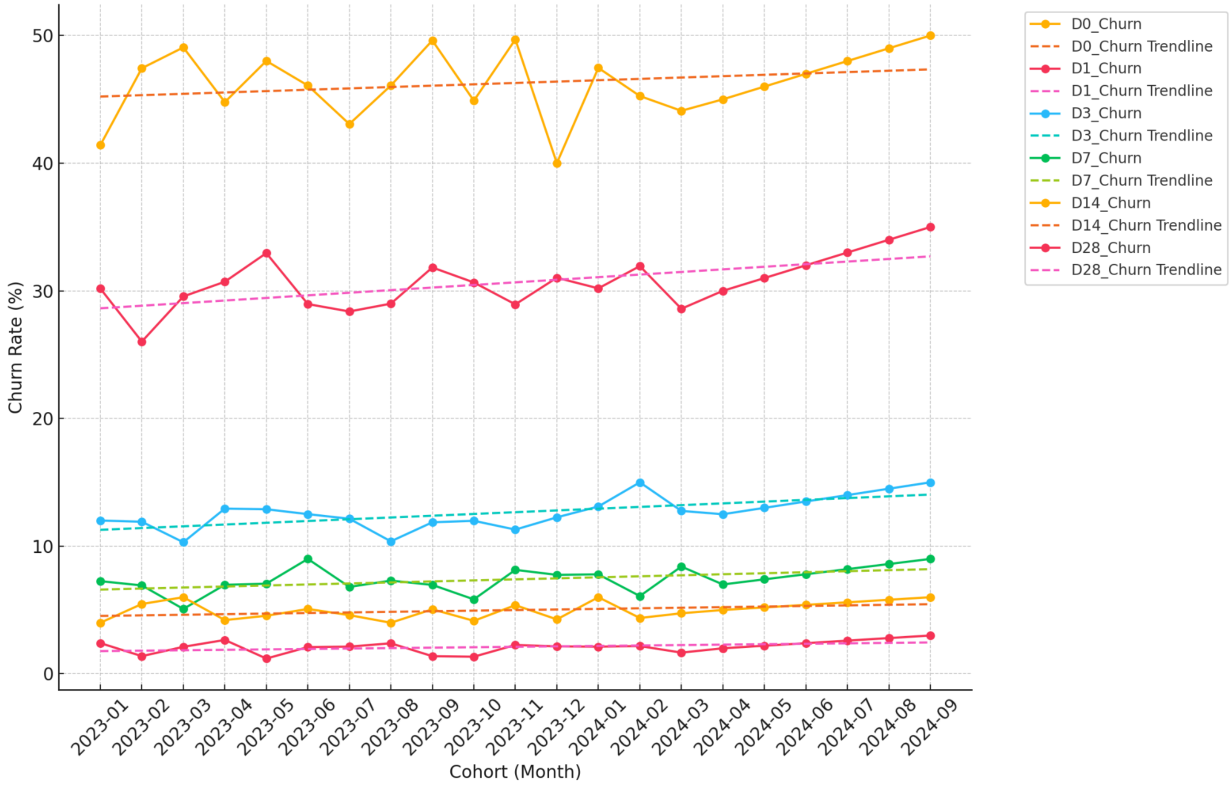
1. **When we see an increasing trend in user churn over the past few months and want to proactively address this issue by identifying at-risk users before they churn, I'd like to propose a framework for developing a predictive model to forecast user churn.Describe the data we would need, the features we would consider, the type of model we might use, and how we would validate the model's effectiveness.**

Before we start creating a model and forecasting churn using the necessary data, it is crucial to analyze the background of the question to create a comprehensive analysis.

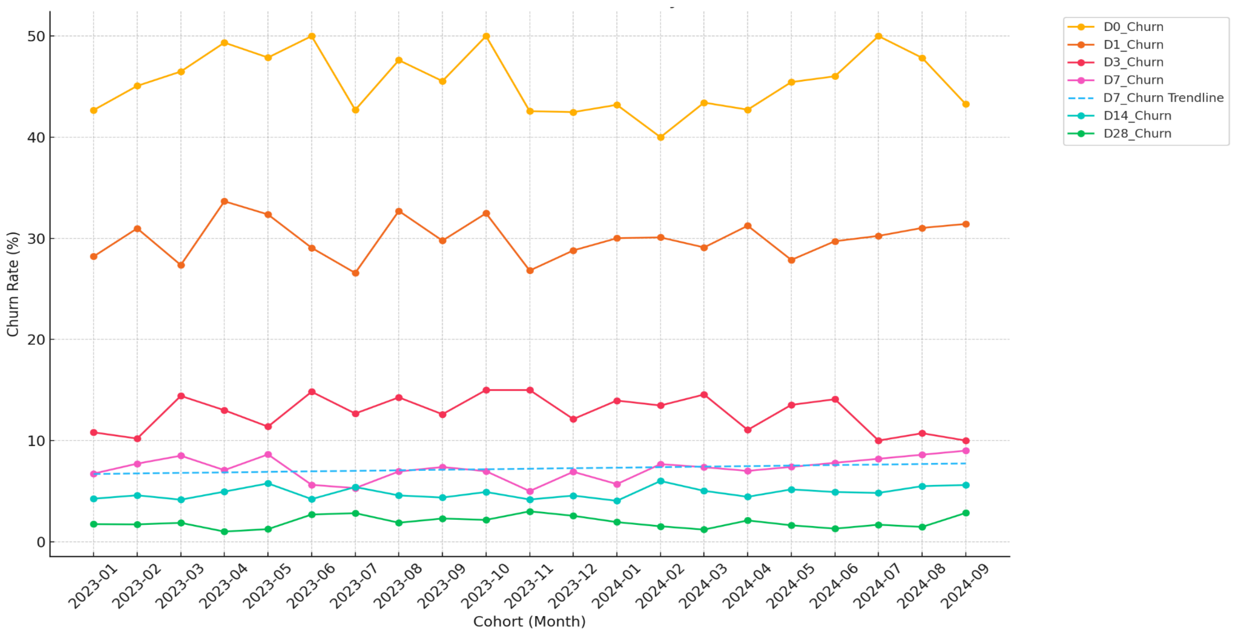
The phrase "Increasing trend in user churn" is a broad term used to identify problems. To properly define and address the issue, we need to ask the right questions.

a. **Is this increase cumulative, or is it related to a specific time horizon?** In other words, are the D0, D1, D3, D7, and D30 churn rates increasing simultaneously, or does an increase in the churn rate at a specific time point (e.g., **DX** churn) drive the overall cumulative churn rate?

If we observe a churn trend like the graph below, it could indicate a different underlying issue. This may not be a periodic problem, as in option **b** (if it were periodic, we would expect to see the same trend as in the previous year), but rather a fundamental problem, as suggested in options **c** or **d**.



If we observe a churn trend like the one shown in the graph below, it suggests a different problem, as we are only seeing an increase in the D7 churn rate in recent months. Therefore, we should focus on analyzing the **D7-related features** in our product. Analyzing churn rates for other days or looking at users outside of this specific timeframe could lead to misleading conclusions.



b. **What data are we comparing in this analysis?**
Did the churn rate increase compared to the same period last year? Did the churn rates for these specific periods also increase last year? Is this a periodic increase, or is it something new?

c. **Have there been any major changes to the product recently?**
For example, has a new feature or event been introduced?

d. **Have there been any recent changes in marketing channels or campaign strategies?**
For instance, has the campaign shifted from a D0 ROAS focused campaign to a D7 BLENDED ROAS campaign?

The examples and questions above are just a few that can help guide our analysis, and many more can be added. I will dive deeper into these points in the subsequent stages.

Ultimately, it is crucial to ask the right and most relevant questions tailored to the specific problem at hand. We must base our answers and models on the insights gained from these questions. Failing to do so could lead to biased estimations, which would increase the likelihood of errors in our model. This, in turn, can result in poor strategic and financial decisions.

2. Data Requirements

For churn analysis, we can divide the data into four main categories:

1. **User Profile**
2. **User Behavioral Data**
3. **Marketing Data**
    - **Campaigns**
    - **Channels**
4. **Event Data (Feature Data)**

User profile

User profile data is one of the most important factors in calculating user churn. Among the profile elements, **country, platform, OS name,** and **app version** are particularly critical for churn analysis.

- Platform (iOS vs. Android) provides a basic segmentation for user analysis. Android offers a broader range of options, while iOS tends to offer more distinct user behavior insights. For example, this difference is often significantly reflected in In-App Purchase (IAP) rates.
- Additionally, when there is a sufficient number of users, the **device model** can also become an important indicator of churn, as it can correlate with user behavior or satisfaction levels.
- Country is another key factor that can greatly influence user segmentation. For instance, the behavior of users in countries like India may differ from users in the U.S. due to cultural, economic, or regional preferences. Identifying these differences is essential for accurate churn analysis.

By considering these user profile elements, we can create more targeted and accurate churn predictions based on specific user segments.

key	**type**	**description**
user_id	varchar	Unique identifier for tracking

os_name	varchar	Ios,android etc.
os_version	varchar	Important for ATT analysis
device_language	varchar	Important for localization
country	varchar	To account for regional behavior patterns
starting_app_version	int	
last_app_version	int	
user_start_ts	timestamp	First time opening app
first_session_id	int	
last_session_id	int	
att_status	boolean	false,true
ip_adress	varchar	Important forIdentifying places after ATT
device_model	varchar	Important for user segmentation

User Behavioral Data

Normally, the table below represents a table that has been created from the raw data tables (such as `datatable_sessions`, `datatable_log`, `datatable_events`, `datatable_iap` etc.) through table operations. The raw data tables contain all the requests from users, and we define the aggregate data by creating a new table for the purpose of generating ad-hoc or daily reports.

key	**type**	**description**
user_id	varchar	Unique identifier for tracking

ab_test		structure	
	is_for_new_user	boolean	
	test_id	int	
	test_name	string	
	test_variant	string	
d1_login		varchar	We can expand these as our preferences.
d3_login		varchar	
d7_login		varchar	
d14_login (etc)		varchar	
total_iap_revenue		int	
total_ad_revenue		int	
total_revenue		boolean	
iap_count		int	
session_count		int	
total_session_duration		varchar	
session_id		varchar	
session_time		int	

Marketing Data

Each marketing campaign targets distinct user demographics due to differences in the channels used, which affects churn rates. For instance, if we are running a **ROAS7** campaign, we expect the **Cost per Install (CPI)** to be higher than with a broad install campaign, indirectly influencing user churn. Similarly, campaigns run on platforms like **TikTok** and **Meta** will yield different retention and churn rates due to the differing audience characteristics on each platform.

key	**type**	**description**
user_id	varchar	Unique identifier for tracking
network_name	string	Important for identify channels
campaign_name	string	Important for growth and marketing
ad_name	string	Interstitial, rewarded, banner etc.
is_organic	boolean	Important for ASO strategy
applovin_id	varchar	
appsflyer_id	varchar	

Event Data (Feature Data)

Actually, this part could also be analyzed within the user’s behavioral data. However, as mentioned earlier, if a new feature that affects the churn rate has been added, it’s a good practice to create a separate data table for it.

Typically, user data related to events is stored in the `datatable_event` table, specifically in the `event_data` column. To focus on a specific event, we can limit the query to the event we’re interested in. For example, the query

`where`

`data_table.event_data.event_id = "specific_event"`

would help us identify and retrieve the event data for the specific event we're analyzing.

key		**type**	**description**
user_id		varchar	Unique identifier for tracking
ab_test		structure	
	is_for_new_user	boolean	
	test_id	int	
	test_name	string	
	test_variant	string	
event_data	*event_id*	varchar	
	event_name	string	
	event_start_date	timesamp	
	event_type	string	
	join_count	int	cumulative
	join_type	string	Main menu, notification ect.
	session_id	varchar	
	session_time	int	
app_name		string	
app_version		varchar	
event_ts		timestamp	
market_name		string	
session_id		varchar	

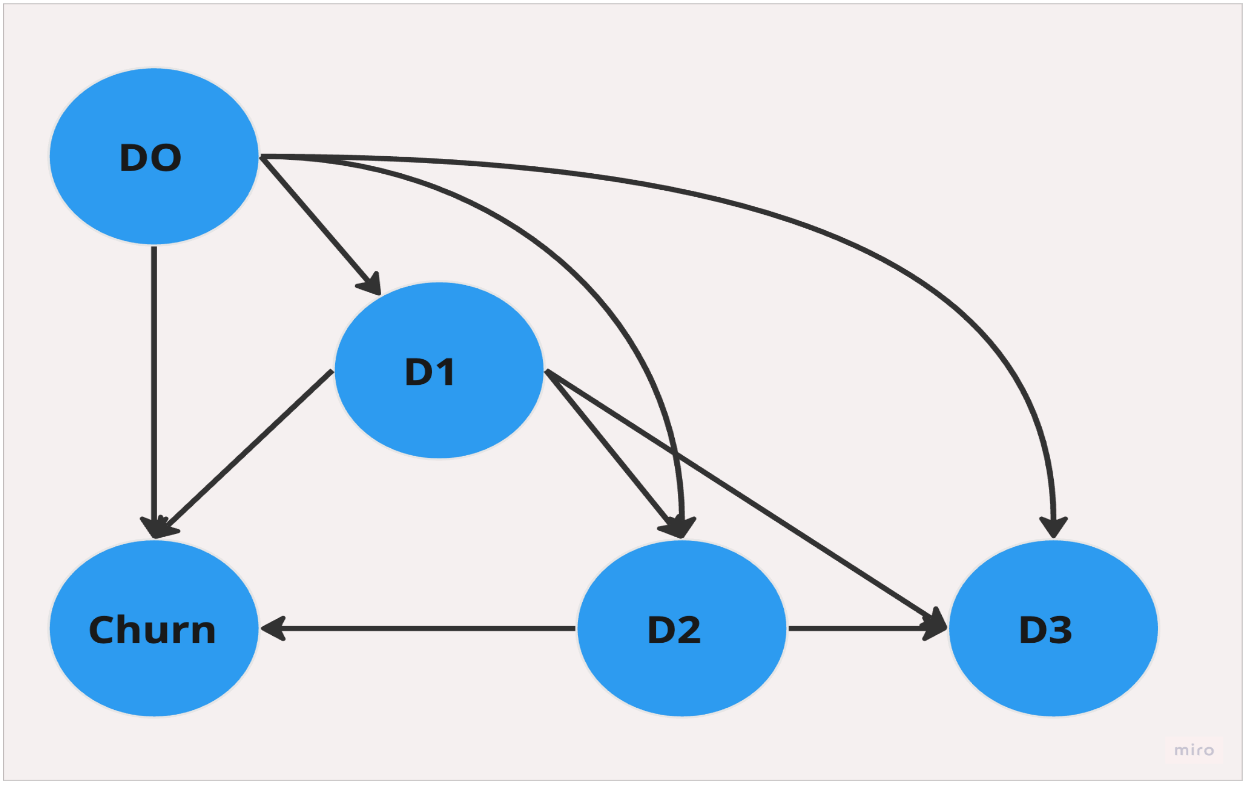
3. Model Selection

I would like to focus on two different models.

- The first one will be a simple, **day-focused model** using **Markov Chain** to analyze the time series data of players.
- The second model will be a **regression analysis** focused on **behavioral data**.

Model-I

The Markov chain to model daily player retention or churn is composed of separate states {0,1,...,n} for each day after install. With this chain we can model what a player is most likely to do next given they just played on their nth day since install



- The Markov chain is represented as a matrix **T** where T[i,j] contains the probability of transition from state i to j.
    - Initialize the elements of T to 0.

- ○ Played on Di then next on Dj without any intervening days.
- ○ T[0,0] means the count of all installs who played on D0 but never played again
- ○ T[0,1] means the count of all installs who next played on D1.

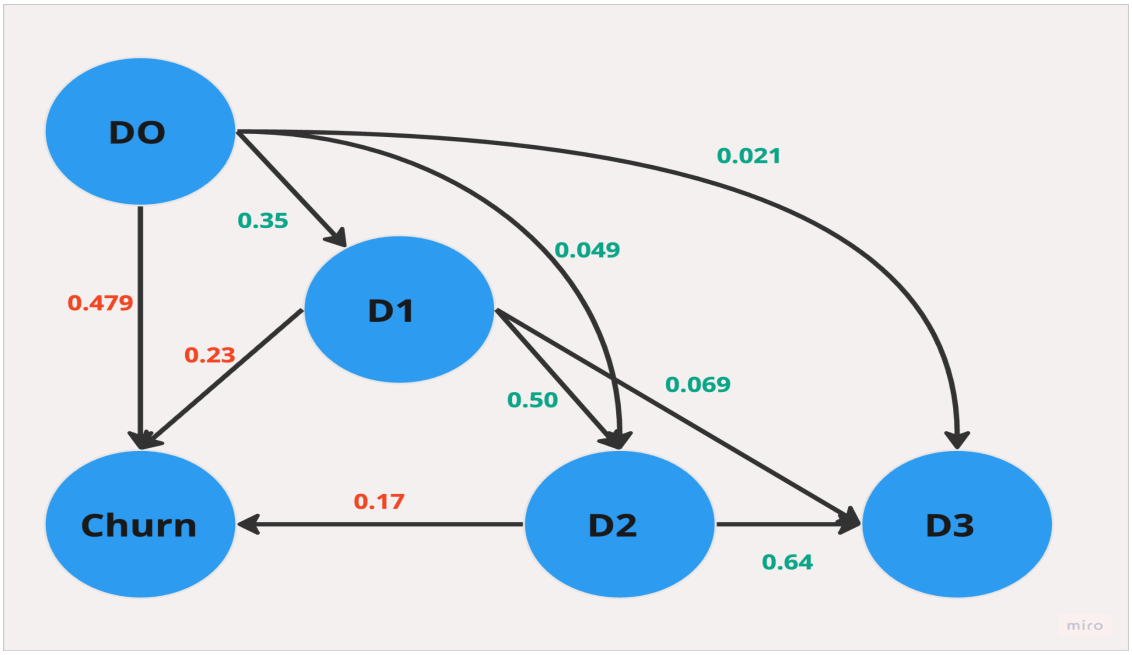
The **3-day period** is used here purely for simplicity in demonstration. As we expand the **nxn matrix**, the combinations of days increase, which in turn increases the computational load of the query. Therefore, the data we choose for analysis becomes crucial.

We can perform a **comparative analysis** by selecting the most reliable data from our historical dataset to ensure that we're working with high-quality information. This approach helps us manage the complexity and focus on the most impactful data points.

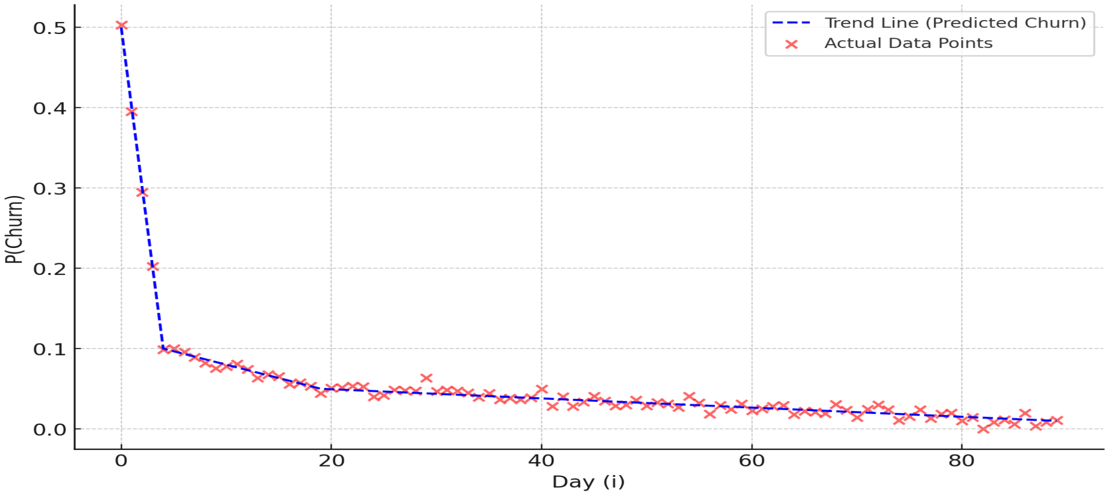
i	**entered**	**churned**	**p_churn**	**p_next**
0	7891	3775	0.47	0.35
1	2929	678	0.23	0.50
2	2120	361	0.17	0.64
3	1769	209	0.11	0.71

i	**j**	**entered**	**players**	**p_next**
0	**1**	7891	2929	0.35
0	**2**	7891	384	0.049
0	**3**	7891	162	0.021
1	**2**	2929	1736	0.50
1	**3**	2929	200	0.069

2	**3**	2120	1407	0.644



At the end of these calculations, the matrix gives probabilities as above.



How to predict user churn with this and trustworthiness?

In this scenario, we want to determine whether the daily churn rates fall within the predicted historical boundaries. For each day:

$x_0 <$ $p\_d_1 <$ x_1
$x_0 <$ $p\_d_2 <$ x_1
$x_0 <$ $p\_d_3 <$ x_1

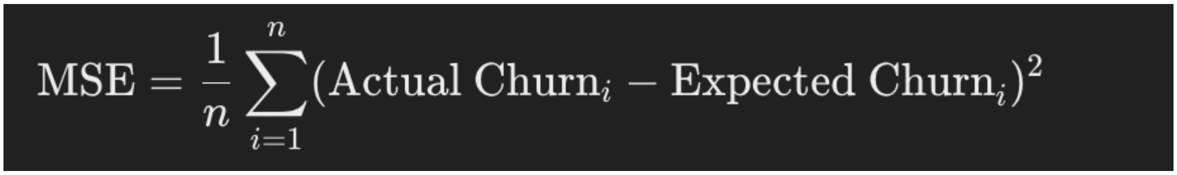
- p_d_x are the actual churn probabilities for Day 1, Day 2, and Day 3.
- x_0 and x_1 are the lower and upper boundaries for the churn rates, derived from historical data.

If the results from the new data fall above these thresholds, it could be a sign of an issue.

For example, if we observe that the active users from the past month show a churn rate outside of these limits, we may need to diagnose and identify the problem. For instance, if the D4 churn rate is unexpectedly high in the new data, we might need to raise an alarm for users in D1, D2, and D3.

Mean Squared Error (MSE)

MSE measures how close the actual churn rates are to the central value of the historical boundaries



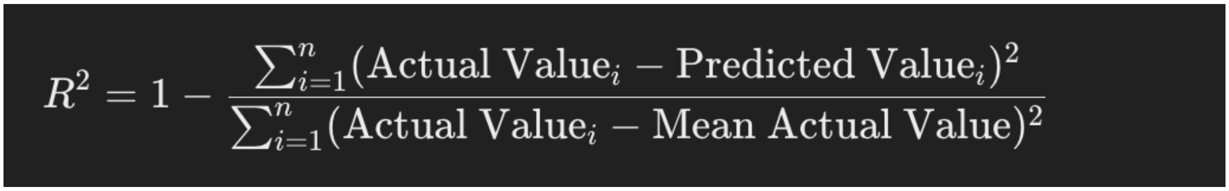
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\text{Actual Churn}_i - \text{Expected Churn}_i)^2$$

For each day iii, calculate the difference between the actual churn and the midpoint of the boundaries.

MSE quantifies how far the actual churn values deviate from the predicted churn trendline. Smaller MSE values indicate a tighter fit, meaning the predictions are closely aligned with the real data.

A low MSE suggests that historical data can reliably predict current churn behavior. If MSE is high, the historical model may not reflect current churn dynamics accurately.

Coefficient of Determination (R^2)



$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\text{Actual Value}_i - \text{Predicted Value}_i)^2}{\sum_{i=1}^{n}(\text{Actual Value}_i - \text{Mean Actual Value})^2}$$

R^2 measures how much of the variance in actual churn data is explained by the model (trendline). A higher R^2 value (closer to 1) indicates that the model accounts for most of the variability.

- **$R^2 \approx 1$**: Excellent fit; historical data explains the current churn behavior well.
- **$R^2 \approx 0$**: Poor fit; historical data doesn’t explain current churn behavior.

Our model is simple taking only one user feature into account. Adding additional features such as session_time, LTV, average_session_length increase the efficacy of the model. We will point it out in model-2.

Model-II

Model-2 will provide us with a forecast that allows for a more detailed interpretation of the user, with multiple explanatory variables. The number of these variables can be selected by us.

In statistics, as we add unknown variables into our equation, the **confidence interval** of the model increases, and the **margin of error** decreases.

However, caution is needed during this process. Adding too many parameters can lead to **biased estimations while explaining dependent variables**, increasing the risk of obtaining incorrect results.

a. In this model, I provide these example variables with historical data

Explanatory variables:

- `sessiontime`
- `ltv`
- `platform`
- `Country_code (tier_rank)`

Dependent variables:

- *Dx_login* (**It means that the last login user day before churn**)

b. Our approach,

- **a regression model** on the historical dataset using features:

  `sessiontime`, `ltv`, `platform`, and `country_code`.

- Predict **churn_day** for users in the new dataset.

c. OLS estimation and checking our data (these are dummy variables. Results are meaningless, just example)

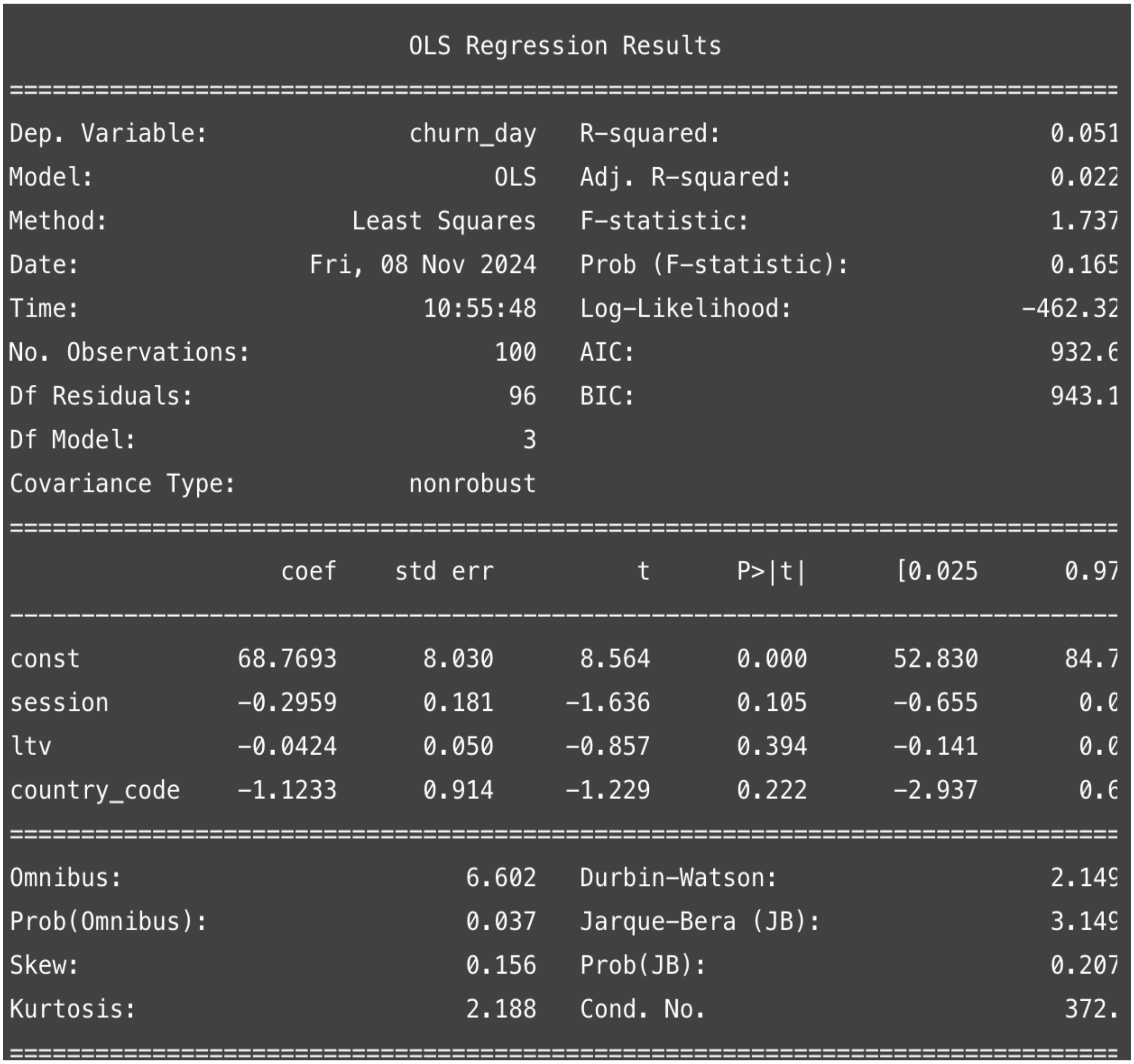
We can use these libraries to analyze our data.

Pandas

numpy

statsmodels.api

matplotlib.pyplot



```
                            OLS Regression Results
==============================================================================
Dep. Variable:              churn_day   R-squared:                       0.051
Model:                            OLS   Adj. R-squared:                  0.022
Method:                 Least Squares   F-statistic:                     1.737
Date:                Fri, 08 Nov 2024   Prob (F-statistic):              0.165
Time:                        10:55:48   Log-Likelihood:                -462.32
No. Observations:                 100   AIC:                             932.6
Df Residuals:                      96   BIC:                             943.1
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                   coef    std err          t      P>|t|      [0.025      0.97
------------------------------------------------------------------------------
const           68.7693      8.030      8.564      0.000      52.830      84.7
session         -0.2959      0.181     -1.636      0.105      -0.655       0.0
ltv             -0.0424      0.050     -0.857      0.394      -0.141       0.0
country_code    -1.1233      0.914     -1.229      0.222      -2.937       0.6
==============================================================================
Omnibus:                        6.602   Durbin-Watson:                   2.149
Prob(Omnibus):                  0.037   Jarque-Bera (JB):                3.149
Skew:                           0.156   Prob(JB):                        0.207
Kurtosis:                       2.188   Cond. No.                         372.
==============================================================================
```

1. **R-squared**:
    - Only about 5.1% of the variance in `churn_day` is explained by the model. This suggests that additional predictors may be needed.
2. **Coefficients**:
    - **Session**: −0.29 suggesting that increased playtime is weakly associated with reduced churn days.
    - **LTV**: −0.042, indicating a minimal relationship between LTV and churn.
    - **Country Code**: −1.12, indicating minor impact on churn prediction.

3. **Assumption Checks**: These are very crucial checks for the trustability and biassenes of our assumption. If E(u) does not equal 0 or there is a homoscedastic assumption we should change our sample.
    - **E(u) = 0**: Residuals' mean is near zero. This is crucial for linear regression
    - **No Heteroscedasticity**: Can be tested via Breusch-Pagan or White tests.
    - **No Autocorrelation**: Durbin-Watson statistic indicates no serious autocorrelation in residuals.

Correlation Matrix: Correlation analysis will help you identify relationships between variables and assess multicollinearity.

Correlation values range from -1 to 1:

- **-1**: Perfect negative correlation.
- **0**: No correlation.
- **1**: Perfect positive correlation.

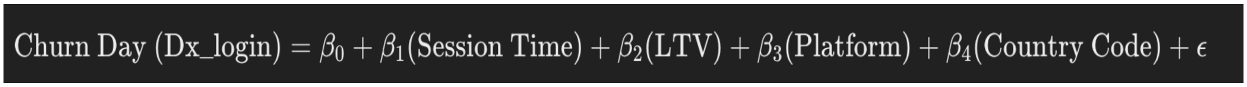
- **Intercept (β_0)**: Baseline churn day when all features are zero.
- **Coefficients (β_1, β_2, β_3, β_4)**: Impact of each predictor on churn day.
- **Error Term (ε)**: Captures unexplained variance.

$$\min \quad SSR = \sum \hat{u}_t^2 = \sum \left(Y_t - \hat{\beta}_0 + \hat{\beta}_1 X_{t1} + \hat{\beta}_2 X_{t2} \right)^2$$

$$\frac{\partial SSR}{\partial \hat{\beta}_0} = \sum (-2) \left(Y_t - \hat{\beta}_0 + \hat{\beta}_1 X_{t1} + \hat{\beta}_2 X_{t2} \right) = 0 \qquad \Rightarrow \sum \hat{u}_t = 0$$

$$\frac{\partial SSR}{\partial \hat{\beta}_1} = \sum (-2) X_{t1} \left(Y_t - \hat{\beta}_0 + \hat{\beta}_1 X_{t1} + \hat{\beta}_2 X_{t2} \right) = 0 \qquad \Rightarrow \sum X_{t1} \hat{u}_t = 0$$

$$\frac{\partial SSR}{\partial \hat{\beta}_2} = \sum (-2) X_{t2} \left(Y_t - \hat{\beta}_0 + \hat{\beta}_1 X_{t1} + \hat{\beta}_2 X_{t2} \right) = 0 \qquad \Rightarrow \sum X_{t2} \hat{u}_t = 0$$



$$\text{Churn Day (Dx\_login)} = \beta_0 + \beta_1(\text{Session Time}) + \beta_2(\text{LTV}) + \beta_3(\text{Platform}) + \beta_4(\text{Country Code}) + \epsilon$$

Explanation of Terms

- **Intercept** : The baseline `churn_day` value when all predictors are zero.
- **Session** : For each additional unit of playtime, how does churn day change?
- **LTV** : A change in LTV how to change churn day is.

Negative Aspects:

Risk of Multicollinearity:

- Correlation between explanatory variables (e.g., LTV and session time) may bias coefficients.

Linear Model Limitations:

- Assumes linear relationships, which may not hold in real-world data.
- May underperform in cases of non-linear dynamics or interactions.

Residual Analysis Dependency:

- Assumption checks (E(u) = 0, homoscedasticity) are crucial. Any violation may invalidate model conclusions.

Positive Aspects:

Multiple Variables for Better Prediction:

- Incorporating **session time**, **LTV**, and **country_code** improves interpretability.
- Helps identify high-risk groups based on multiple criteria.

Interpretability:

- OLS regression provides clear coefficients, allowing easy explanation of each variable's impact on churn.

Assumption Testing:

- Verifying assumptions (e.g., no heteroscedasticity, no autocorrelation) increases the reliability of your model.