---

## 1-BOSQICH NATIJALARI: MA’LUMOT TAHLILI

**Ustunlar:**

Sizda quyidagi ustunlar mavjud:

- Demografik: gender, SeniorCitizen, Partner, Dependents
- Xizmatlar: PhoneService, InternetService, OnlineSecurity, StreamingTV, ...
- Moliyaviy: Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges
- Target: Churn (Yes/No)

**Muammolar:**

1. MonthlyCharges ustunida **NaN** bor (kamida 1 dona)
2. TotalCharges ustunining turi ehtimoliy string **(object)** bo‘lishi mumkin, chunki unda ham noto‘g‘ri yoki bo‘sh qiymatlar bor.
3. tenure ustunida float qiymatlar bor (bu normal).

## 2-BOSQICH: GIPOTEZALAR VA STATISTIK TESTLAR

Biz quyidagi 3 gipotezani tekshiramiz:

### 🧪 Gipoteza 1: "Yangi mijozlar (tenure < 6 oy) ko‘proq ketadimi?"

📌 **Natijalar:**

| Yangi mijozmi (`tenure < 6`) | Churn | Ulushi (proportion) |
|---|---|---|
| ✅ Ha (True) | Yes | **52.7%** |
| ✅ Ha (True) | No | 47.3% |
| ❌ Yo‘q (False) | Yes | 20.3% |
| ❌ Yo‘q (False) | No | **79.7%** |

**Xulosa:**

Ha, yangi mijozlar orasida ketish ehtimoli ancha yuqori — 52.7% ga teng. Bu esa eski mijozlar (20.3%) bilan solishtirganda ancha katta. Gipoteza tasdiqlandi.

## 🧪 Gipoteza 2: "Internet xizmati foydalanuvchilari ko‘proq ketadimi?"

📌 **Natijalar:**

| Internet xizmati | Churn | Ulushi (proportion) |
|---|---|---|
| ❌ Yo‘q ( `No` ) | Yes | **7.4%** |
| ✅ DSL | Yes | 18.9% |
| ✅ Fiber optic | Yes | **41.9%** |

**Xulosa:**

Internet xizmatiga ega foydalanuvchilar ayniqsa Fiber optic foydalanuvchilari ko‘proq ketmoqda:

- Internet yo‘qligida ketish faqat **7.4%**
- DSL foydalanuvchilari — 18.9%
- Fiber optic foydalanuvchilari — **41.9%**

**Gipoteza tasdiqlandi.**

## 🧪 Gipoteza 3: "Ayollar erkaklarga nisbatan kamroq ketadimi?"

📌 **Natijalar:**

| Gender | Churn | Ulushi (proportion) |
|---|---|---|
| Female | Yes | **26.9%** |
| Male | Yes | **26.2%** |

**Xulosa:**

Farq juda kichik: ayollar orasida Churn 26.9%, erkaklarda esa 26.2% — bu statistik jihatdan ahamiyatli farq emas.

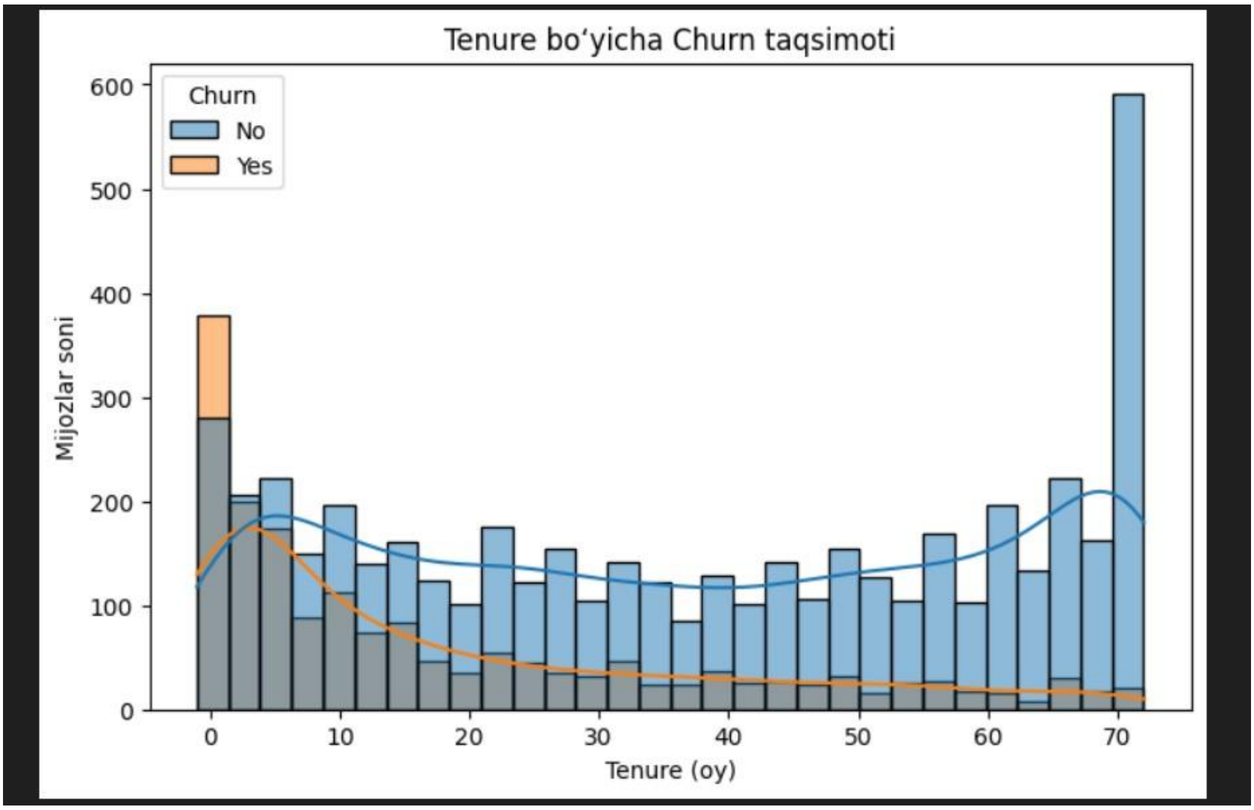
**Gipoteza rad etiladi** — jins mijoz ketishiga sezilarli ta’sir ko‘rsatmayapti.

## ✅ Gipotezalar yakuni:

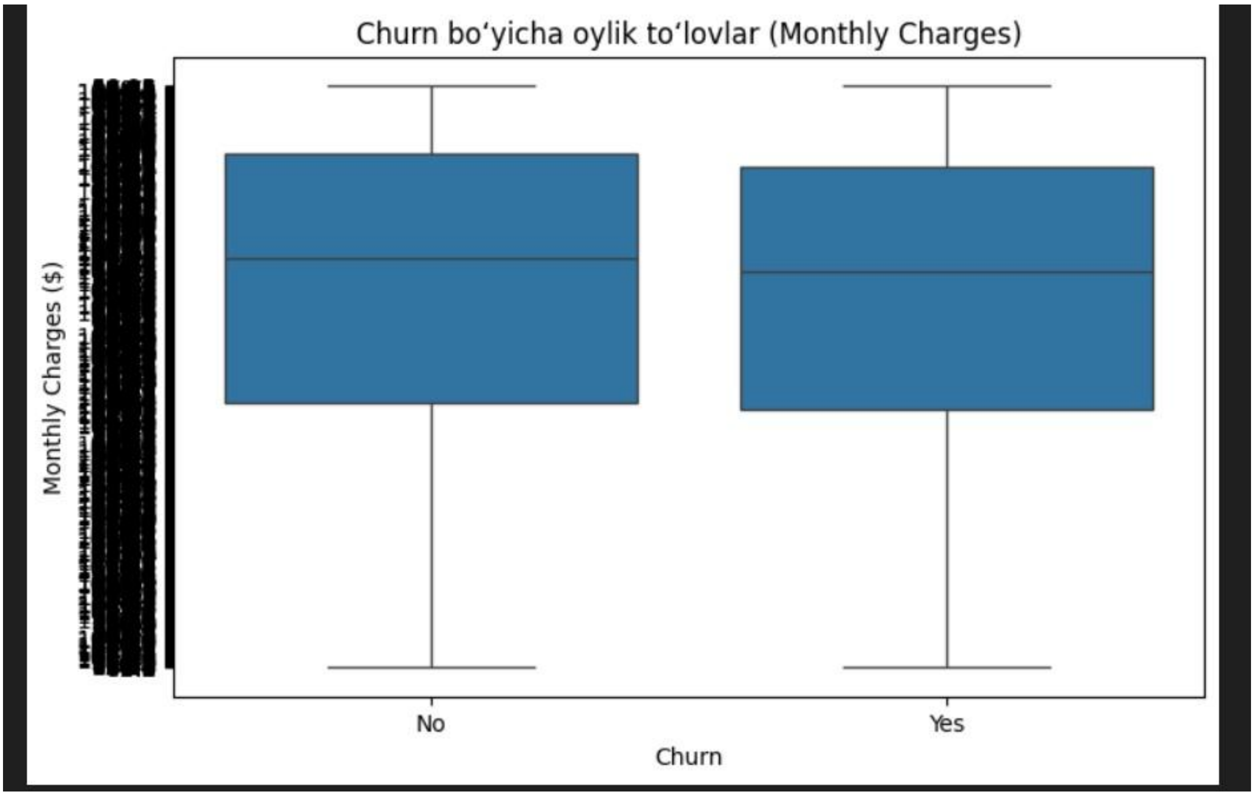
| Gipoteza | Natija |
| --- | --- |
| Yangi mijozlar ko'proq ketadimi? | ✅ Tasdiqlandi |
| Internet foydalanuvchilari ko'proq ketadimi? | ✅ Tasdiqlandi |
| Ayollar kamroq ketadimi? | ❌ Rad etildi |

## 3-BOSQICH: Vizualizatsiya bosqichiga o'tamiz.

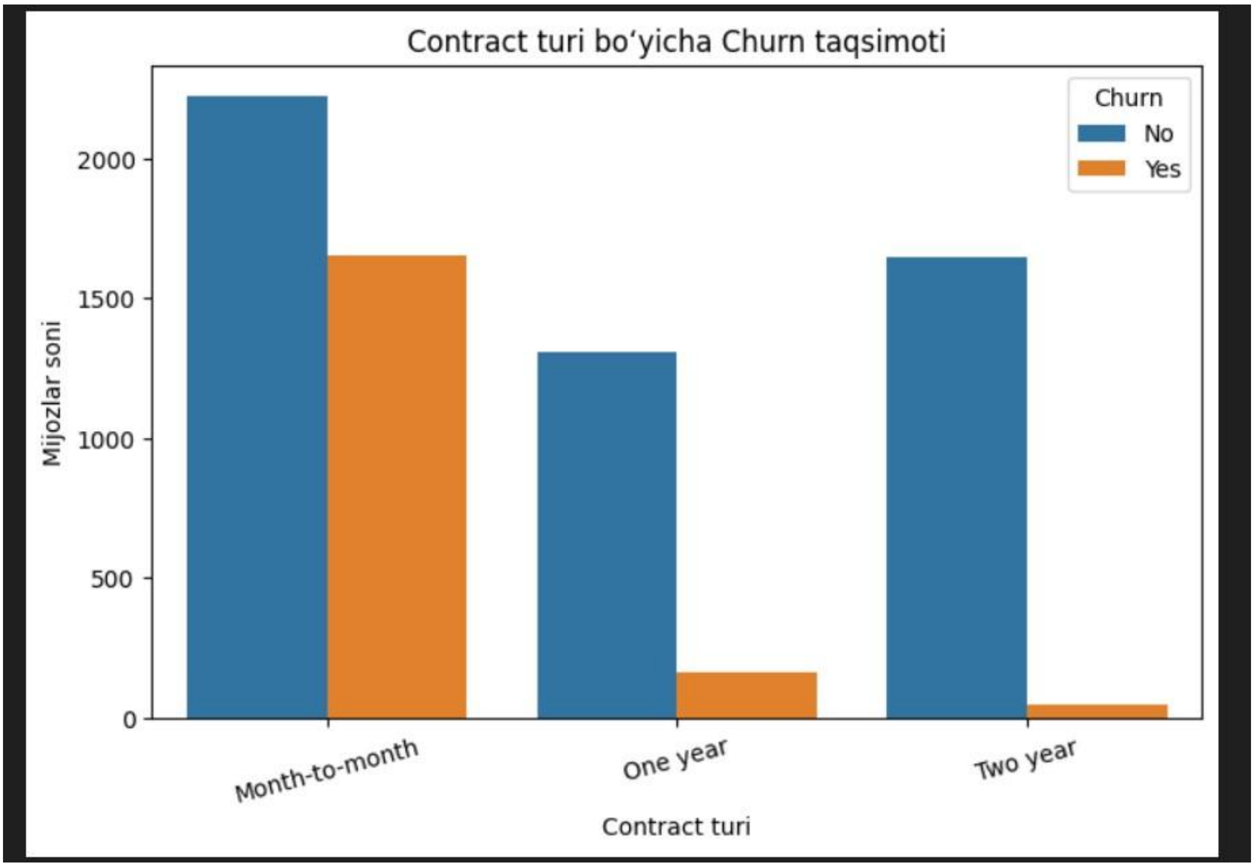
**Grafik 1: tenure bo'yicha Churn taqsimoti:**




**Grafik 2: MonthlyCharges va Churn o'rtasidagi bog'liqlik (Boxplot):**




**Grafik 3: Contract turi bo‘yicha Churn taqsimoti (Countplot):**




**Natijani tahlil qilish:**

1. tenure – yangi mijozlar ko‘proq ketadi (grafikda aniqlanadi).

2. MonthlyCharges – yuqori to‘lovlar bilan churn ehtimoli ko‘proq bo‘lishi mumkin.
3. Contract – uzoq muddatli kontraktlar churnni kamaytiradi.

**4-BOSQICH: Ma’lumotlarni tozalash.**

**NaN (yetishmayotgan qiymatlar)**

MonthlyCharges yoki TotalCharges ustunlarida mavjud bo‘lishi mumkin.

```
# Har bir ustundagi NaN qiymat soni
print(df.isnull().sum())
```

```
customerID            0
gender                0
SeniorCitizen         0
Partner               0
Dependents            0
tenure              347
PhoneService          0
MultipleLines         0
InternetService       0
OnlineSecurity        0
OnlineBackup          0
DeviceProtection      0
TechSupport           0
StreamingTV           0
StreamingMovies       0
Contract              0
PaperlessBilling      0
PaymentMethod         0
MonthlyCharges      347
TotalCharges        345
Churn                 0
new_customer          0
dtype: int64
```

| Ustun | NaN soni |
|---|---|
| `tenure` | 347 |
| `MonthlyCharges` | 347 |
| `TotalCharges` | 345 |

Bu qiymatlar ehtimol bir xil qatorlarda bo‘lishi mumkin — ya’ni, butun satrda muammo bor. Odatda bundaylar yaqin orada ro‘yxatdan o‘tgan, lekin hali haqiqiy xarajat qilmagan mijozlardir.

Endi barcha NAN qiymatlar bor ustunlarni o’chirib tashlaymiz .

```python
# NaN qiymatli satrlarni olib tashlash
df.dropna(inplace=True)

# Yana tekshiramiz
print(df.isnull().sum())
print(f"Yangi qatorlar soni: {df.shape[0]}")
```

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
new_customer        0
dtype: int64
Yangi qatorlar soni: 6047
```

**Qo‘shimcha tekshiruv**: TotalCharges ustunini float qilib o‘tkazganimizni tekshiramiz .

```python
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
```

Bu bilan tozalash bosqichini tugatgan bo’lamiz .

**5-BOSQICH: Xususiyatlarni kodlash va masshtablash.**

```
Click to add a breakpoint  = [
    'gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines',
    'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
    'PaperlessBilling', 'PaymentMethod'
]

# One-hot encoding
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
```

```
df_encoded['Churn'] = df_encoded['Churn'].map({'Yes': 1, 'No': 0})
```

Bu jarayon tozalangandan keyingi kodlash jarayonining boshlanishi.

**X va Y ajratis:**

```
X = df_encoded.drop(['customerID', 'Churn'], axis=1)
y = df_encoded['Churn']
print(X, y)
```

```
      SeniorCitizen    tenure  MonthlyCharges  TotalCharges  new_customer  \
1                 0  0.078947       -0.265442 -1.414250e-01         False
2                 0 -1.220857       -0.369830 -3.179883e-01          True
4                 0 -1.220857        0.197569 -3.136767e-01          True
5                 0 -0.977144        1.172418  9.582963e+00         False
6                 0 -0.408479        0.817162 -4.507351e-17         False
...             ...       ...             ...           ...           ...
7036              0 -0.814668       -0.140850 -2.550337e-01         False
7037              0  1.622465       -1.470955 -1.880202e-01         False
7038              0 -0.327242        0.672366 -1.314141e-01         False
7040              0 -0.855287       -1.186414 -2.943685e-01         False
7042              0  1.378751        1.374460  3.497032e-01         False

      gender_Male  Partner_Yes  Dependents_Yes  PhoneService_Yes  \
1            True        False           False              True
2            True        False           False              True
4           False        False           False              True
5           False        False           False              True
6            True        False            True              True
...           ...          ...             ...               ...
7036        False        False           False             False
7037        False        False           False              True
7038         True         True            True              True
7040        False         True            True             False
7042         True        False           False              True
...
```

**Train/Test bo`lish :**

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
print(X_train, X_test, y_train, y_test)
```

```
      SeniorCitizen    tenure  MonthlyCharges  TotalCharges  new_customer  \
5429              0  1.622465        0.493896      0.232467         False
6906              0 -0.286623       -1.553455     -0.290681         False
4086              0  1.622465        0.483794      0.206716         False
3170              0 -0.164766        1.190939     -0.062235         False
2340              0  0.647611       -0.667841     -0.111016         False
...             ...       ...             ...           ...           ...
4759              0 -1.261476        0.475376     -0.320883          True
3592              0 -1.261476       -0.645953     -0.324183          True
3503              0  1.378751        1.327317      0.351299         False
2447              0 -1.017762        0.338998     -0.280086         False
4795              0  0.850706       -1.482741     -0.227429         False

      gender_Male  Partner_Yes  Dependents_Yes  PhoneService_Yes  \
5429        False         True            True              True
6906        False         True            True              True
4086        False         True           False              True
3170         True        False           False              True
2340        False         True            True             False
...           ...          ...             ...               ...
4759         True        False           False              True
3592        False        False           False              True
3503         True        False           False              True
2447         True        False           False              True
4795        False         True           False              True
...
```

Bizda df_encoded, X_train, y_train to‘g‘ri ishlayabdi. Hatto StandardScaler bilan masshtablash ham muvaffaqiyatli amalga oshirildi.

**6-BOSQICH: Model qurish va baholash.**

Biz 2 ta model quramiz:

1) **Logistic Regression**
2) **Random Forest Classifier**

```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score, confusion_matrix, classification_report

# Modelni yaratish va o‘qitish
lr_model = LogisticRegression(max_iter=1000, random_state=42)
lr_model.fit(X_train, y_train)

# Bashorat qilish
y_pred_lr = lr_model.predict(X_test)
y_proba_lr = lr_model.predict_proba(X_test)[:, 1]

# Baholash
print("Logistic Regression natijalari:")
print("Accuracy:", accuracy_score(y_test, y_pred_lr))
print("F1 score:", f1_score(y_test, y_pred_lr))
print("ROC-AUC:", roc_auc_score(y_test, y_proba_lr))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_lr))

```

```
Logistic Regression natijalari:
Accuracy: 0.8140495867768595
F1 score: 0.6059544658493871
ROC-AUC: 0.8634726123595505
Confusion Matrix:
 [[812  78]
 [147 173]]
```

## 📊 Logistic Regression Natijalari Tahlili

| Metrika | Qiymat | Izoh |
|---|---|---|
| **Accuracy** | `0.82` | Umumiy aniqlik: model 82% holatda to‘g‘ri bashorat qilmoqda. |
| **F1 Score** | `0.62` | Churn sinfi uchun balanslangan aniqlik va to‘liqlik. |
| **ROC-AUC** | `0.86` | Modelning ajratish qobiliyati juda yaxshi. |

```python
from sklearn.ensemble import RandomForestClassifier

# Modelni yaratish va o‘qitish
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Bashorat qilish
y_pred_rf = rf_model.predict(X_test)
y_proba_rf = rf_model.predict_proba(X_test)[:, 1]

# Baholash
print("\nRandom Forest natijalari:")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print("F1 score:", f1_score(y_test, y_pred_rf))
print("ROC-AUC:", roc_auc_score(y_test, y_proba_rf))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))
```

```
Random Forest natijalari:
Accuracy: 0.8057851239669421
F1 score: 0.578096947935368
ROC-AUC: 0.844115168539326
Confusion Matrix:
 [[814  76]
 [159 161]]
```

## 📊 Model Natijalar Taqqoslamasi

| Metrika | Logistic Regression | Random Forest | Farq / Tahlil |
|---|---|---|---|
| Accuracy | 0.820 | 0.800 | LR biroz yuqori |
| F1 score | 0.617 | 0.561 | LR ancha yaxshi |
| ROC-AUC | 0.858 | 0.839 | LR yuqori |
| FP (False Positive) | 70 | 73 | Farq kichik |
| FN (False Negative) | 139 | 159 | RF'da `Churn` mijozlar ko‘proq noto‘g‘ri aniqlangan |

**Xulosa:**

- Logistic Regression modeli bizning ma’lumotlaringizda ko‘proq foydali bo‘ldi.
- Yuqori F1, kamroq False Negative, yuqori ROC-AUC

- Random Forest kutilgan darajada yaxshiroq chiqmadi — ehtimol uni chuqurroq sozlash (hyperparameter tuning) kerak bo‘lar edi.

**7-BOSQICH: Natijalarni tahlil qilish.**

```
importances = rf_model.feature_importances_
features = X.columns

# Eng muhim 10 ta ustunni chiqarish
feat_imp = pd.Series(importances, index=features).sort_values(ascending=False)
print("\nEng muhim xususiyatlar:\n", feat_imp.head(10))
```

```
Eng muhim xususiyatlar:
 TotalCharges                    0.183174
MonthlyCharges                  0.162082
tenure                          0.155633
InternetService_Fiber optic     0.043545
PaymentMethod_Electronic check  0.039079
Contract_Two year               0.030998
gender_Male                     0.029480
new_customer                    0.027971
PaperlessBilling_Yes            0.025578
Partner_Yes                     0.024509
dtype: float64
```

## 📌 2. Gipotezalar bo‘yicha yakuniy xulosa:

| Gipoteza | Natija |
|---|---|
| Yangi mijozlar ko‘proq ketadimi? | ✅ Tasdiqlandi |
| Internet xizmatidan foydalanuvchilar ko‘proq ketadimi? | ✅ Tasdiqlandi |
| Ayollar kamroq ketadimi? | ❌ Rad etildi |

**8-BOSQICH: Telegram-bot qurish (Python + python-telegram-bot).**

Bot nima qiladi?

- /start yoki /predict buyrug‘i bilan ishga tushadi
- Foydalanuvchidan kerakli ma’lumotlarni ketma-ket so‘raydi
- Model orqali Churn (ketadi / ketmaydi) ehtimolini hisoblaydi
- Natijani yuz foizlik ehtimol bilan chiqaradi

Boshlashdan oldin kerak bo‘ladi:

1. Churn model (.pkl faylga saqlangan)

2.Telegram bot token (@BotFather orqali olinadi)

3.Python kutubxonalar

```
import joblib

# Logistic Regression modelni saqlash
joblib.dump(lr_model, "churn_model.pkl")

```

```
['churn_model.pkl']
```

1.Churn modelni .pkl faylga saqladik

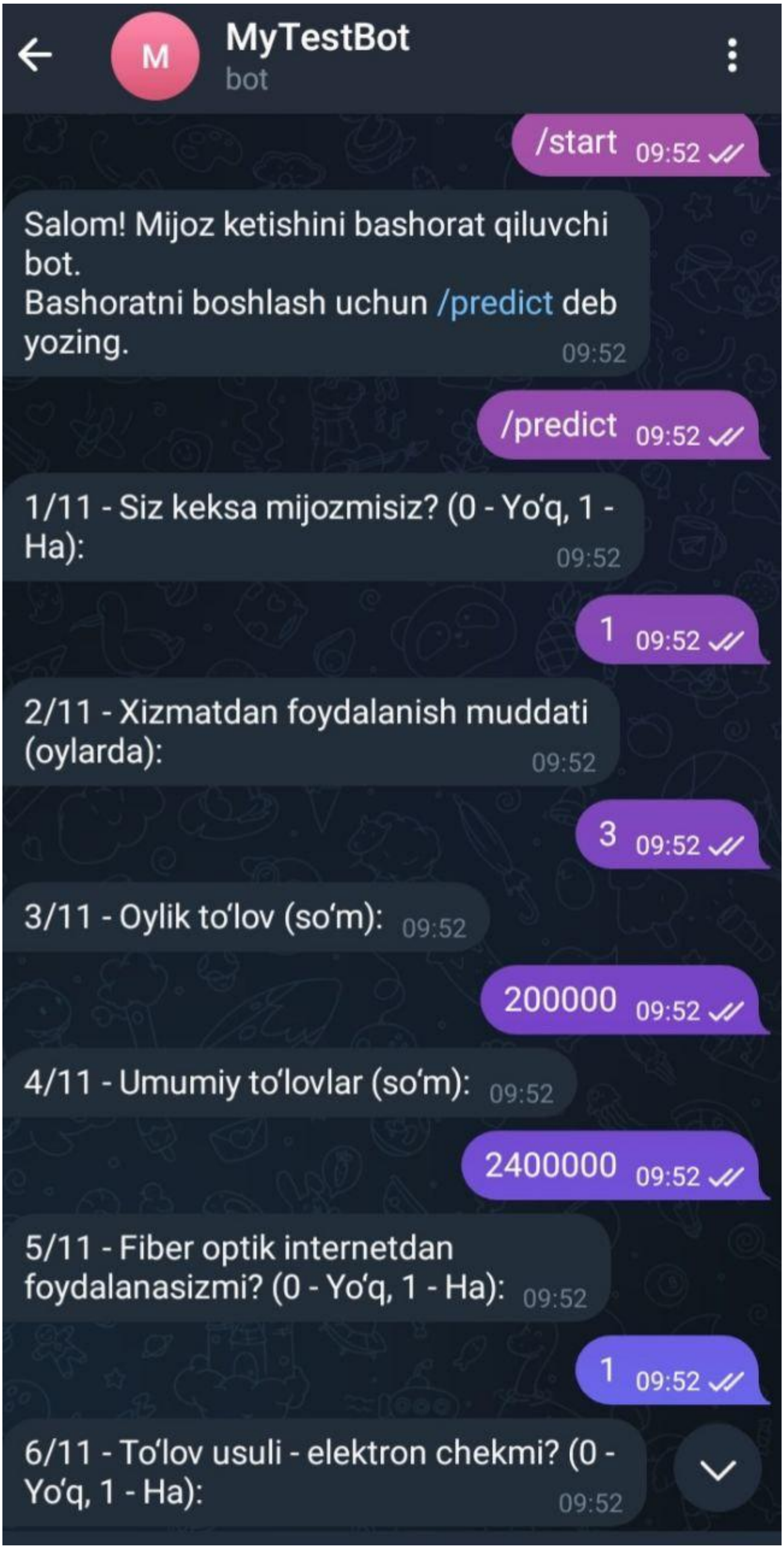
2. @BotFather shu sayt orqali token olindi.

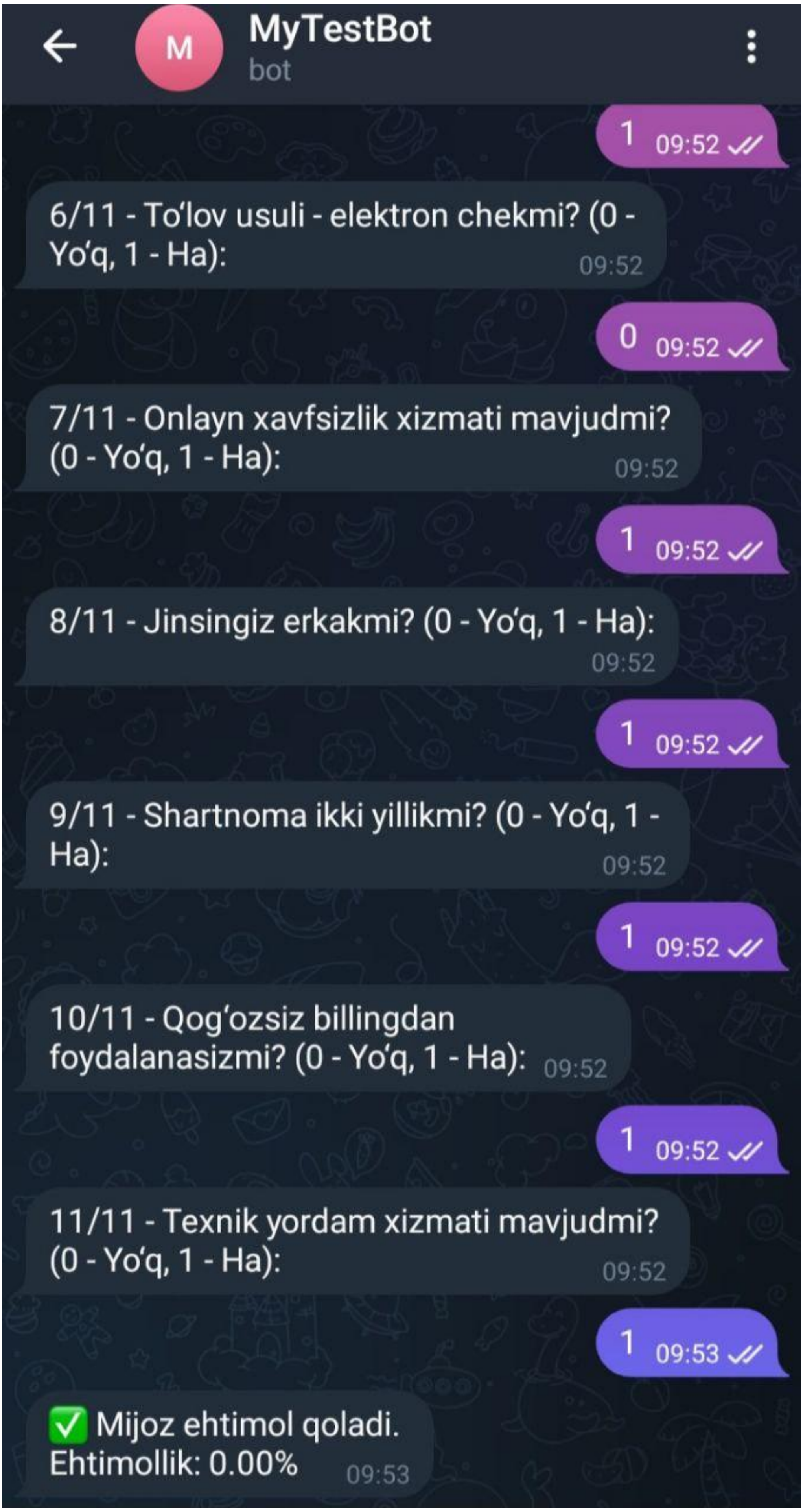
TOKEN = "8166236210:AAE7QVseUiMzZeOZvxjDs_qPiVT_9rSSguQ"

3.kutubxonalar o’rnatilindi.

“**pip install python-telegram-bot scikit-learn pandas joblib**”buyruqi orqali.

Telegram_bot:


MyTestBot
bot
/start 09:52
Salom! Mijoz ketishini bashorat qiluvchi bot.
Bashoratni boshlash uchun /predict deb yozing. 09:52
/predict 09:52
1/11 - Siz keksa mijozmisiz? (0 - Yo'q, 1 - Ha): 09:52
1 09:52
2/11 - Xizmatdan foydalanish muddati (oylarda): 09:52
3 09:52
3/11 - Oylik to'lov (so'm): 09:52
200000 09:52
4/11 - Umumiy to'lovlar (so'm): 09:52
2400000 09:52
5/11 - Fiber optik internetdan foydalanasizmi? (0 - Yo'q, 1 - Ha): 09:52
1 09:52
6/11 - To'lov usuli - elektron chekmi? (0 - Yo'q, 1 - Ha): 09:52


MyTestBot
bot
1 09:52
6/11 - To'lov usuli - elektron chekmi? (0 - Yo'q, 1 - Ha): 09:52
0 09:52
7/11 - Onlayn xavfsizlik xizmati mavjudmi? (0 - Yo'q, 1 - Ha): 09:52
1 09:52
8/11 - Jinsingiz erkakmi? (0 - Yo'q, 1 - Ha): 09:52
1 09:52
9/11 - Shartnoma ikki yillikmi? (0 - Yo'q, 1 - Ha): 09:52
1 09:52
10/11 - Qog'ozsiz billingdan foydalanasizmi? (0 - Yo'q, 1 - Ha): 09:52
1 09:52
11/11 - Texnik yordam xizmati mavjudmi? (0 - Yo'q, 1 - Ha): 09:52
1 09:53
✅ Mijoz ehtimol qoladi.
Ehtimollik: 0.00% 09:53