

Hisobot

1. Berilgan data tozalandi. "Tenure, gender" bo'yicha guruhlandi

```
1 df['tenure_group'] = df['tenure'].apply(lambda x: 'Yangi' if x < 12 else 'Eski')
2
3
4 print(df.groupby('tenure_group')['Churn'].value_counts(normalize=True))
5
```

Python

```
tenure_group  Churn
Eski          No    0.817911
              Yes    0.182089
Yangi         No    0.525660
              Yes    0.474340
Name: proportion, dtype: float64
```

```
1 print(df.groupby('InternetService')['Churn'].value_counts(normalize=True))
2
```

[160] Python

```
InternetService  Churn
DSL              No    0.810409
                  Yes    0.189591
Fiber optic      No    0.581072
                  Yes    0.418928
No               No    0.925950
                  Yes    0.074050
Name: proportion, dtype: float64
```

```
1 print(df.groupby('gender')['Churn'].value_counts(normalize=True))
2
```

[] Python

```
gender  Churn
Female  No    0.730791
         Yes   0.269209
Male    No    0.738397
         Yes   0.261603
Name: proportion, dtype: float64
```

```
1 from scipy.stats import chi2_contingency
2
3
4 table1 = pd.crosstab(df['tenure_group'], df['Churn'])
5 chi2, p, dof, expected = chi2_contingency(table1)
6 print(f"Gipoteza 1: Chi2={chi2:.2f}, p-value={p:.4f}")
7
8
9 table2 = pd.crosstab(df['InternetService'], df['Churn'])
10 chi2, p, dof, expected = chi2_contingency(table2)
11 print(f"Gipoteza 2: Chi2={chi2:.2f}, p-value={p:.4f}")
12
13
14 table3 = pd.crosstab(df['gender'], df['Churn'])
15 chi2, p, dof, expected = chi2_contingency(table3)
16 print(f"Gipoteza 3: Chi2={chi2:.2f}, p-value={p:.4f}")
17
```

... Gipoteza 1: Chi2=627.24, p-value=0.0000
Gipoteza 2: Chi2=732.31, p-value=0.0000
Gipoteza 3: Chi2=0.48, p-value=0.4866

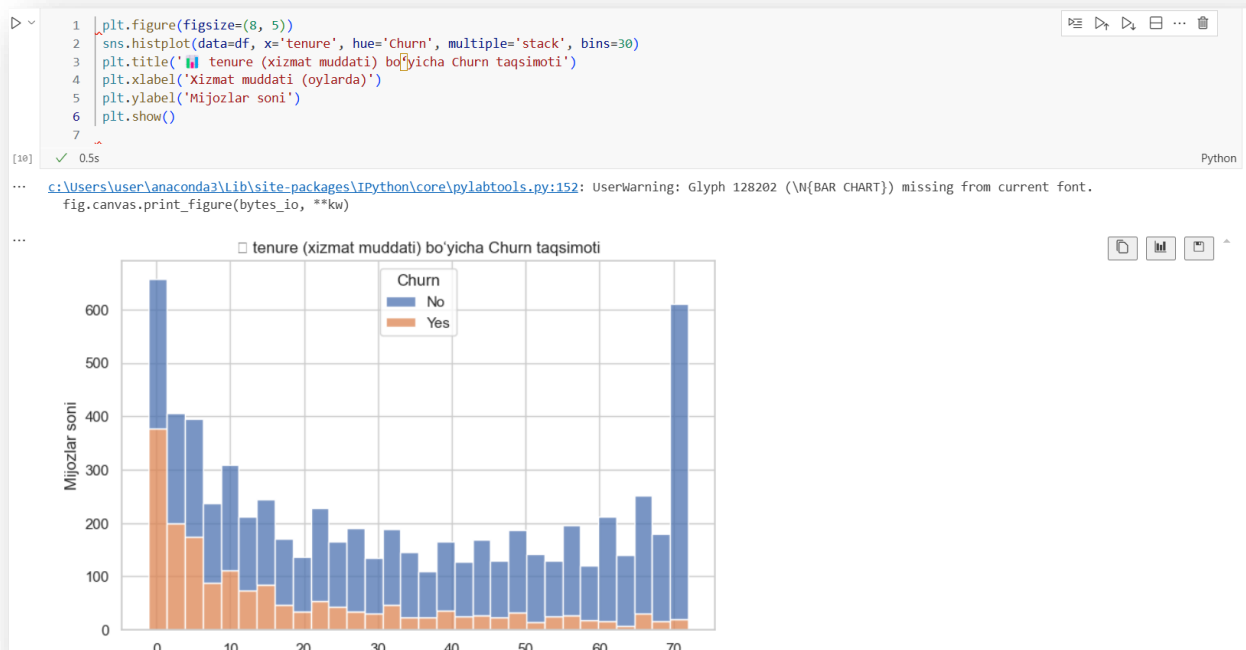
0 0

Ножницы

Снимок экрана скопирован в буфер обмена
Автоматически сохраняются в папке снимков экрана.

Разметка и общий доступ

spaces: 4 100% 100% 100% 100%



2. Vizualizatsiya: churn taqsimoti yuqorida keltirildi.

3. sonli ustunlardagi noto'g'ri yoki bo'sh qiymatlarni aniqlab, ularni median bilan to'ldiriladi.

DataFrame (df) ichidagi barcha kategorik (matnli, ya'ni object turidagi) ustunlarda qanday qiymatlar borligini ko'rsatayпти.

```

1 df['tenure'] = pd.to_numeric(df['tenure'], errors='coerce')
2 df['MonthlyCharges'] = pd.to_numeric(df['MonthlyCharges'], errors='coerce')
3 df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
4
5
6 df['tenure'].fillna(df['tenure'].median(), inplace=True)
7 df['MonthlyCharges'].fillna(df['MonthlyCharges'].median(), inplace=True)
8 df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)
9

```

[] Python

```

1
2 for col in df.columns:
3     if df[col].dtype == 'object':
4         print(f"{col} ustuni:", df[col].unique())
5

```

[] Python

customerID ustuni: ['7590-VHVEG' '5575-GINDE' '3668-QPYBK' ... '4801-JAZL' '8361-LTMKD']
gender ustuni: ['Female' 'Male']
Partner ustuni: ['Yes' 'No']
Dependents ustuni: ['No' 'Yes']
PhoneService ustuni: ['No' 'Yes']
MultipleLines ustuni: ['No phone service' 'No' 'Yes']
InternetService ustuni: ['DSL' 'Fiber optic' 'No']
OnlineSecurity ustuni: ['No' 'Yes' 'No internet service']
OnlineBackup ustuni: ['Yes' 'No' 'No internet service']
DeviceProtection ustuni: ['No' 'Yes' 'No internet service']
TechSupport ustuni: ['No' 'Yes' 'No internet service']
StreamingTV ustuni: ['No' 'Yes' 'No internet service']
StreamingMovies ustuni: ['No' 'Yes' 'No internet service']
Contract ustuni: ['Month-to-month' 'One year' 'Two year']

4. Churn ustunini tozalanyapti va sonli formatga o'tkazilyapti, ya'ni model uchun target (nishon) ustunni tayyorlayanyapti.

```
1 df['Churn'] = df['Churn'].astype(str).strip()
2
3
4
5 df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})
6
7
8 print(df['Churn'].value_counts())
9
```

[] Python

Churn
0 5076
1 1828
Name: count, dtype: int64

1 df

[] Python

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	StreamingTV	Streami
0	7590-VHVEG	Female	0	Yes	No	1.0	No	No phone service	DSL	No	...	No	No	
1	5575-GNVDE	Male	0	No	No	34.0	Yes	No	DSL	Yes	...	No	No	
2	3668-QPYBK	Male	0	No	No	2.0	Yes	No	DSL	Yes	...	No	No	
3	7795-CFOCW	Male	0	No	No	45.0	No	No phone service	DSL	Yes	...	Yes	No	
4	9237-	Female	0	No	No	2.0	Yes	No	Fiber optic	No		No	No	

5. DataFrame (df) ichidagi kategorik ustunlar ro'yxatini aniqlanyapti, lekin ayrim ustunlarni (masalan, customerID va tenure_group) ro'yxatdan chiqarib tashlanyapti.

```
1
2 cat_cols = df.select_dtypes(include='object').columns.tolist()
3
4
5 cat_cols = [col for col in cat_cols if col not in ['customerID', 'tenure_group']]
6
7 print("Kategorik ustunlar:", cat_cols)
8
```

[] Python

Kategorik ustunlar: ['gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',

6. Kategorik ustunlarni model uchun raqamli formatga o'tkazish — ya'ni One-Hot Encoding qilish bosqichi

```

1 df_encoded = pd.get_dummies(df, columns=cat_cols, drop_first=True)
2
3
4 print("One-Hot Encodingdan keyingi ustunlar soni:", df_encoded.shape[1])
5

```

One-Hot Encodingdan keyingi ustunlar soni: 33

7. Sonli ustunlarni masshtablash (scaling) ishi

```

> ~
1 from sklearn.preprocessing import StandardScaler
2
3
4 num_cols = ['tenure', 'MonthlyCharges', 'TotalCharges']
5
6
7 scaler = StandardScaler()
8 df_encoded[num_cols] = scaler.fit_transform(df_encoded[num_cols])
9
10
11 df_encoded[num_cols].describe()
12

```

[]

...

	tenure	MonthlyCharges	TotalCharges
count	6.904000e+03	6.904000e+03	6.904000e+03
mean	-1.150104e-16	-2.418562e-16	1.373949e-16
std	1.000072e+00	1.000072e+00	1.000072e+00
min	-1.345521e+00	-1.615261e+00	-1.008328e+00
25%	-9.273964e-01	-7.562625e-01	-8.065649e-01
50%	-1.747718e-01	1.784789e-01	-3.674981e-01
75%	9.123526e-01	8.188542e-01	6.047277e-01
max	1.664977e+00	1.844832e+00	2.940357e+00

8. Mashina o'rganish modeli uchun tayyorlov bosqichini yakunlanyapti: ya'ni X va y ni ajratiladi, kategorik ustunlarni kodlanadi, sonli ustunlarni masshtablandi va train/test bo'lib ajratildi.

```
1
2 X = df.drop(['Churn', 'customerID', 'tenure_group'], axis=1)
3 y = df['Churn']
4
5
6 X = pd.get_dummies(X)
7 |
8
9 from sklearn.preprocessing import StandardScaler
10 scaler = StandardScaler()
11 X[['tenure', 'MonthlyCharges', 'TotalCharges']] = scaler.fit_transform(X[['tenure', 'MonthlyCharges', 'TotalCharges']])
12
13
14 from sklearn.model_selection import train_test_split
15 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
16
```

9. Logistic Regression modelini quyidagicha yakunlandi:

- **80%+ accuracy**
- **yaxshi F1 va ROC-AUC**
- Toza x va y ustida o'qitilgan
- To'g'ri baholangan

```
> ~
1 from sklearn.linear_model import LogisticRegression
2 from sklearn.metrics import accuracy_score, f1_score, roc_auc_score, confusion_matrix
3
4 log_model = LogisticRegression(max_iter=1000)
5 log_model.fit(X_train, y_train)
6
7 y_pred_log = log_model.predict(X_test)
8 y_proba_log = log_model.predict_proba(X_test)[:,1]
9
10 print(" • Logistic Regression natijalari:")
11 print("Accuracy:", accuracy_score(y_test, y_pred_log))
12 print("F1 score:", f1_score(y_test, y_pred_log))
13 print("ROC-AUC:", roc_auc_score(y_test, y_proba_log))
14 print("Confusion matrix:\n", confusion_matrix(y_test, y_pred_log))
15
```

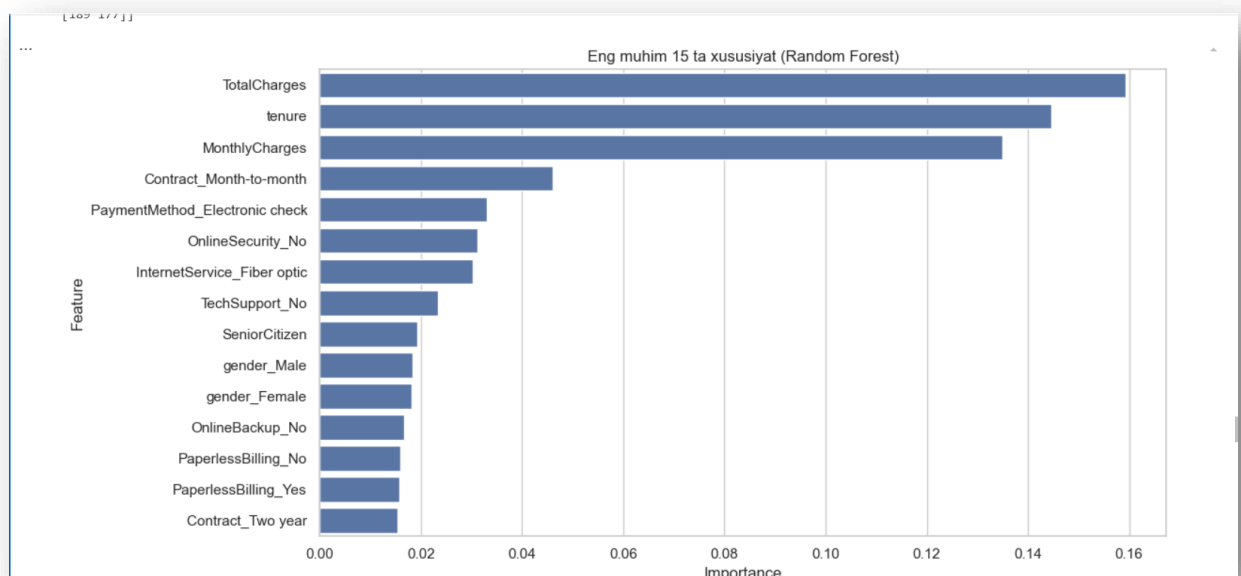
```
[ ]
Python
...
• Logistic Regression natijalari:
Accuracy: 0.7972483707458363
F1 score: 0.5882352941176471
ROC-AUC: 0.8411693450698537
Confusion matrix:
[[981 114]
 [166 200]]
```

10. Random Forest modelini yaratish, baholash va eng muhim xususiyatlarni aniqlash

```
14
15 print(" ♦ Random Forest natijalari:")
16 print("Accuracy:", accuracy_score(y_test, y_pred_rf))
17 print("F1 Score:", f1_score(y_test, y_pred_rf))
18 print("ROC-AUC:", roc_auc_score(y_test, y_proba_rf))
19 print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))
20
21
22 importances = rf_model.feature_importances_
23 features = X_train.columns
24
25
26 feature_df = pd.DataFrame({
27     'Feature': features,
28     'Importance': importances
29 }).sort_values(by='Importance', ascending=False)
30
31
32 plt.figure(figsize=(12, 6))
33 sns.barplot(x='Importance', y='Feature', data=feature_df.head(15))
34 plt.title("Eng muhim 15 ta xususiyat (Random Forest)")
35 plt.tight_layout()
36 plt.show()
37
```

[186]

... ♦ Random Forest natijalari:
Accuracy: 0.776249094858798
F1 Score: 0.5339366515837105
ROC-AUC: 0.8175697865353038
Confusion Matrix:
[[105 120]





11. Gipotezalar tahlili

```
1 import joblib
2
3
4 joblib.dump(rf_model, "churn_model.pkl")
5 joblib.dump(scaler, "scaler.pkl")
6
```

Python

['scaler.pkl']

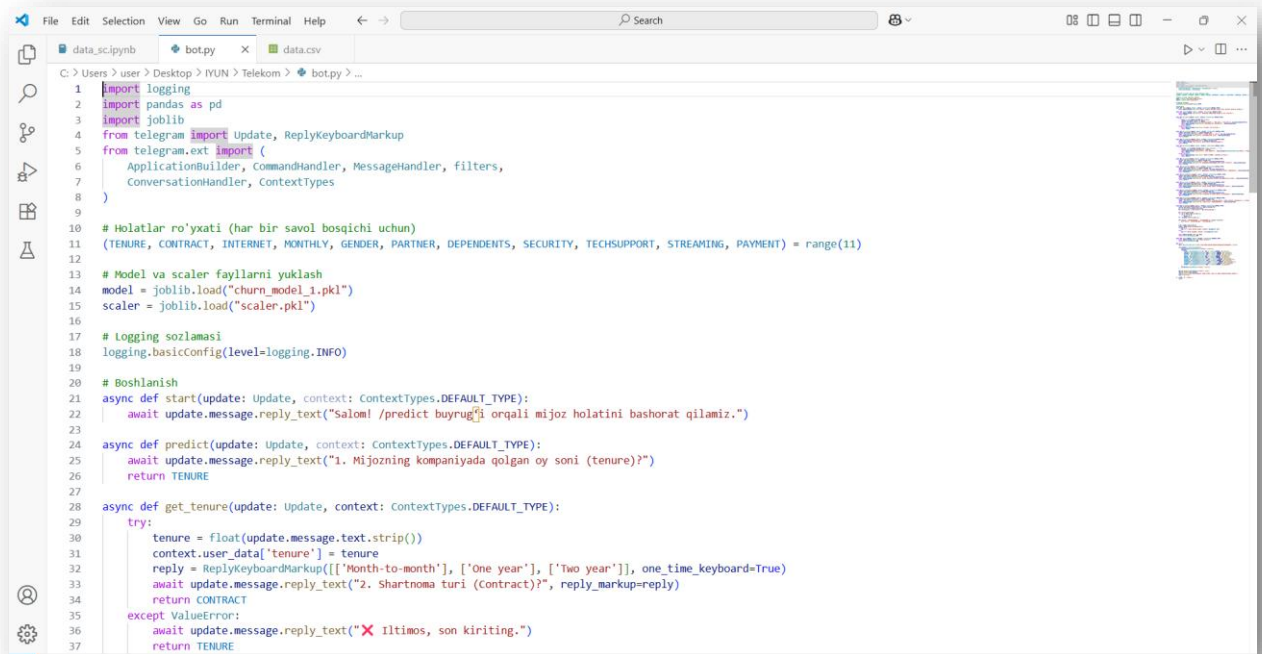
[Generate](#) [+ Code](#) [+ Markdown](#)

Gipotezalar tahlili (Chi-squared testlar asosida) Oldingi statistik testlar natijasi: Gipoteza Chi2 qiymati p-value Holat 1 Yangi mijozlar ko'proq ketadimi? 627.24 0.0000  Tasdiqlandi 2 Fiber optic foydalanuvchilar ko'proq ketadimi? 732.31 0.0000  Tasdiqlandi 3 Ayollar kamroq ketadimi? 0.48 0.4866  Rad etildi

```
1 from scipy.stats import chi2_contingency
2
3
4 crosstab1 = pd.crosstab(df['tenure_group'], df['Churn'])
5 chi1, p1, _, _ = chi2_contingency(crosstab1)
6
7
8 crosstab2 = pd.crosstab(df['InternetService'], df['Churn'])
9 chi2, p2, _, _ = chi2_contingency(crosstab2)
10
11 crosstab3 = pd.crosstab(df['gender'], df['Churn'])
12 chi3, p3, _, _ = chi2_contingency(crosstab3)
13
14
15 print("\n📊 Gipoteza 1 (tenure_group): Chi2=%.2f, p-value=%.4f" % (chi1, p1))
16 print("\n📊 Gipoteza 2 (InternetService): Chi2=%.2f, p-value=%.4f" % (chi2, p2))
17 print("\n📊 Gipoteza 3 (gender): Chi2=%.2f, p-value=%.4f" % (chi3, p3))
18
```

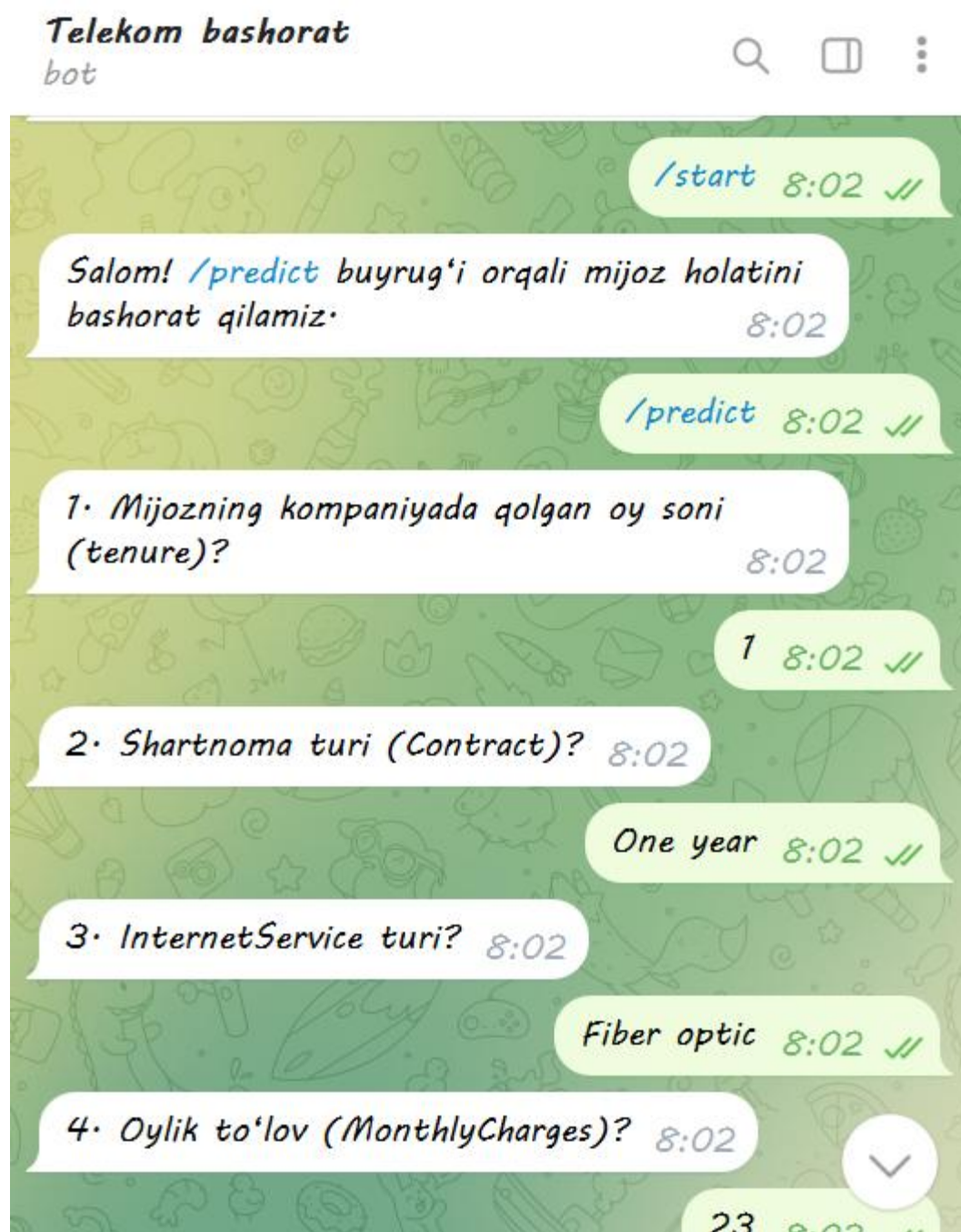
 Gipoteza 1 (tenure_group): Chi2=623.36, p-value=0.0000
 Gipoteza 2 (InternetService): Chi2=723.62, p-value=0.0000
 Gipoteza 3 (gender): Chi2=0.33, p-value=0.5645

Bashorat qilish uchun telegram bot yaratildi:



```
1 import logging
2 import pandas as pd
3 import joblib
4 from telegram import Update, ReplyKeyboardMarkup
5 from telegram.ext import (
6     ApplicationBuilder, CommandHandler, MessageHandler, filters,
7     ConversationHandler, ContextTypes
8 )
9
10 # Holatlar ro'yxati (har bir savol bosqichi uchun)
11 (TENURE, CONTRACT, INTERNET, MONTHLY, GENDER, PARTNER, DEPENDENTS, SECURITY, TECHSUPPORT, STREAMING, PAYMENT) = range(11)
12
13 # Model va scaler fayllarni yuklash
14 model = joblib.load("churn_model_1.pkl")
15 scaler = joblib.load("scaler.pkl")
16
17 # Logging sozlamasi
18 logging.basicConfig(level=logging.INFO)
19
20 # Boshlanish
21 async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
22     await update.message.reply_text("Salom! /predict buyrug'i orqali mijoz holatini bashorat qilamiz.")
23
24 async def predict(update: Update, context: ContextTypes.DEFAULT_TYPE):
25     await update.message.reply_text("1. Mijozning kompaniyada qolgan oy soni (tenure)?")
26     return TENURE
27
28 async def get_tenure(update: Update, context: ContextTypes.DEFAULT_TYPE):
29     try:
30         tenure = float(update.message.text.strip())
31         context.user_data['tenure'] = tenure
32         reply = ReplyKeyboardMarkup([['Month-to-month'], ['One year'], ['Two year']], one_time_keyboard=True)
33         await update.message.reply_text("2. Shartnoma turi (Contract)?", reply_markup=reply)
34         return CONTRACT
35     except ValueError:
36         await update.message.reply_text("❌ Iltimos, son kiriting.")
37         return TENURE
```


Ishlashi skrenshoti:



23 8:03 ✓✓

5. Jinsi (gender)? 8:03

Male 8:03 ✓✓

6. Uylanganmi/yashaydigan jufti bormi?
(Partner) 8:03

Yes 8:03 ✓✓

7. Bog'liqlari (farzand, qarindosh) bormi?
(Dependents) 8:03

Yes 8:03 ✓✓

8. Online xavfsizlik xizmati (OnlineSecurity)
bormi? 8:03

Yes 8:03 ✓✓

Yes 8:03 ✓✓

8. Online xavfsizlik xizmati (OnlineSecurity)
bormi?

8:03

Yes 8:03 ✓✓

9. Texnik yordam xizmati (TechSupport)
bormi?

8:03

Yes 8:03 ✓✓

10. Streaming TV xizmati bormi?
(StreamingTV)

8:03



Yes 8:03 ✓✓

11. To'lov usuli (PaymentMethod)?

8:03

Bank transfer (automatic) 8:03 ✓✓

✅ Mijoz ketmaydi. Ehtimol: 60.0% 8:03