Telekom kompaniyasidagi mijozlar ketishini tahlil qilish va bashorat qilish.

1.Biznes muammosini tushunish

Telekom kompaniyalari uchun mijoz ketishini (churn) bashorat qilish moliyaviy va strategik ahamiyatga ega. Yangi mijoz jalb qilish saqlab qolishdan 5-10 baravar qimmat, ketish esa doimiy daromadni yo‘qotadi. Bashoratli tahlil muammolarni erta aniqlab, chegirma yoki xizmat yaxshilash kabi choralar ko‘rishga yordam beradi. Segmentatsiya xavfli mijozlarga shaxsiy takliflar yo‘naltirish imkonini beradi, bu sadoqatni oshiradi.Xavfli mijozlar qatoriga tenure qisqa (<6 oy) yangi mijozlar, month-to-month shartnomalilar, yuqori to‘lovchilar, fiber optic internet foydalanuvchilari va texnik muammolarga duch kelganlar kiradi. Bularning ketish ehtimoli yuqori bo‘lib, ularga alohida e’tibor talab qilinadi.

2. Ma’lumotlarni tahlil qilish va gipotezalarni tekshirish:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...

Endi bizga berilgan ma’lumotning shakli va o’lchamini kurib chiqamiz:

```
1  df.shape  # Ma'lumotlar to'plamining o'lchamini ko'rsatish
```

```
(7043, 21)
```

End bizga berilgan ma’lumotlar asosida ma’lumotlarni tahlilini ya’ni gipotezalarni ko’rib chiqamiz:

1. Yangi mijozlar ko‘proq ketadimi?

```python
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind, chi2_contingency
# Gipoteza 1: Yangi mijozlar ko‘proq ketadimi?
tenure_churn = df[df['Churn'] == 'Yes']['tenure']
tenure_no_churn = df[df['Churn'] == 'No']['tenure']

print("\nGipoteza 1: Yangi mijozlar ko‘proq ketadimi?")
print(f"Ketganlar o'rtacha tenure: {tenure_churn.mean():.2f}")
print(f"Qolganlar o'rtacha tenure: {tenure_no_churn.mean():.2f}")

t_stat, p_val = ttest_ind(tenure_churn, tenure_no_churn, equal_var=False)
print(f"T-test natijasi: t-statistic = {t_stat:.3f}, p-value = {p_val:.3f}")
if p_val < 0.05:
    print("Natija: Gipoteza tasdiqlandi (farq statistik jihatdan muhim).")
else:
    print("Natija: Gipoteza rad qilindi (farq statistik jihatdan muhim emas).")
```

```
Gipoteza 1: Yangi mijozlar ko‘proq ketadimi?
Ketganlar o'rtacha tenure: 17.98
Qolganlar o'rtacha tenure: 37.57
T-test natijasi: t-statistic = -34.824, p-value = 0.000
Natija: Gipoteza tasdiqlandi (farq statistik jihatdan muhim).
```

2. Internet xizmatidan foydalanuvchilar ko‘proq ketadimi?

```python
# Gipoteza 2: Internet xizmatidan foydalanuvchilar ko‘proq ketadimi?
# Internet xizmatidan foydalanish (Yes/No) va churn (Yes/No) o'rtasidagi bog'liqlikni Chi-kvadrat testi bilan tekshirish

print("\nGipoteza 2: Internet xizmatidan foydalanuvchilar ko‘proq ketadimi?")
contingency_table = pd.crosstab(df['InternetService'], df['Churn'])
print("Kontingens jadvali:")
print(contingency_table)

chi2, p, dof, expected = chi2_contingency(contingency_table)
print(f"Chi-square test natijasi: chi2 = {chi2:.3f}, p-value = {p:.3f}")

if p < 0.05:
    print("Natija: Gipoteza tasdiqlandi (Internet xizmatidan foydalanish va churn o'rtasida bog'liqlik bor).")
else:
    print("Natija: Gipoteza rad qilindi (bog'liqlik aniqlanmadi).")
```

Python

```
Gipoteza 2: Internet xizmatidan foydalanuvchilar ko‘proq ketadimi?
Kontingens jadvali:
Churn              No   Yes
InternetService
DSL              1962   459
Fiber optic      1799  1297
No               1413   113
Chi-square test natijasi: chi2 = 732.310, p-value = 0.000
Natija: Gipoteza tasdiqlandi (Internet xizmatidan foydalanish va churn o'rtasida bog'liqlik bor).
```

3.Ayollar kamroq ketadimi?

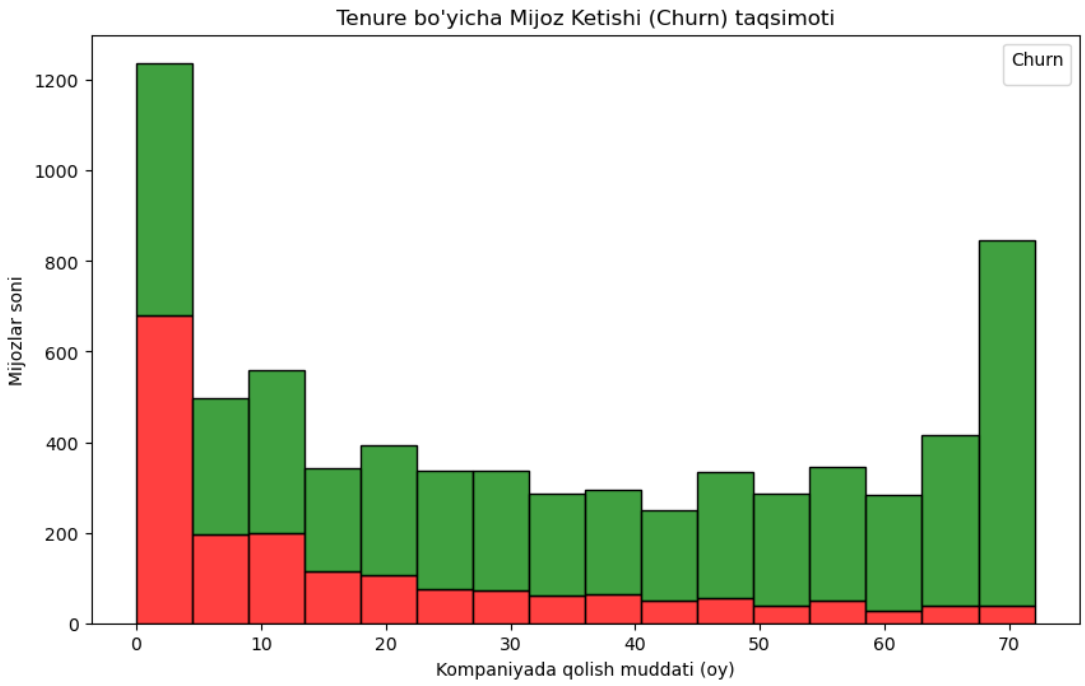
```python
# Gipoteza 3: Ayollar kamroq ketadimi?
# Jins (gender) va churn o'rtasidagi farqni Chi-kvadrat testi bilan tekshirish

print("\nGipoteza 3: Ayollar kamroq ketadimi?")
contingency_gender = pd.crosstab(df['gender'], df['Churn'])
print("Kontingens jadvali:")
print(contingency_gender)

chi2_gender, p_gender, dof_gender, expected_gender = chi2_contingency(contingency_gender)
print(f"Chi-square test natijasi: chi2 = {chi2_gender:.3f}, p-value = {p_gender:.3f}")

if p_gender < 0.05:
    print("Natija: Gipoteza tasdiqlandi (jins va churn o'rtasida farq bor).")
else:
    print("Natija: Gipoteza rad qilindi (jins va churn o'rtasida farq yo'q).")
```

```
Gipoteza 3: Ayollar kamroq ketadimi?
Kontingens jadvali:
Churn     No  Yes
gender
Female  2549  939
Male    2625  930
Chi-square test natijasi: chi2 = 0.484, p-value = 0.487
Natija: Gipoteza rad qilindi (jins va churn o'rtasida farq yo'q).
```

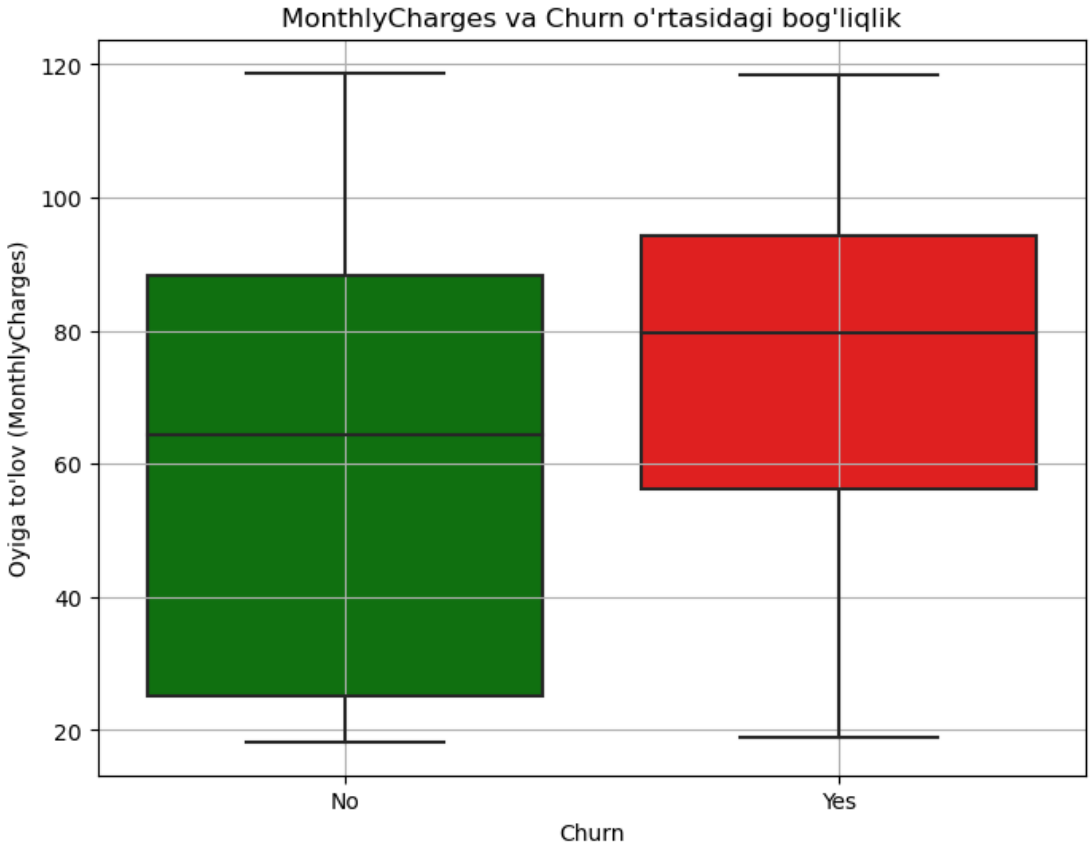
Bu natijalarni olganmizdan kiyingi bosqichga o’tamiz.

3. Vizualizatsiya:

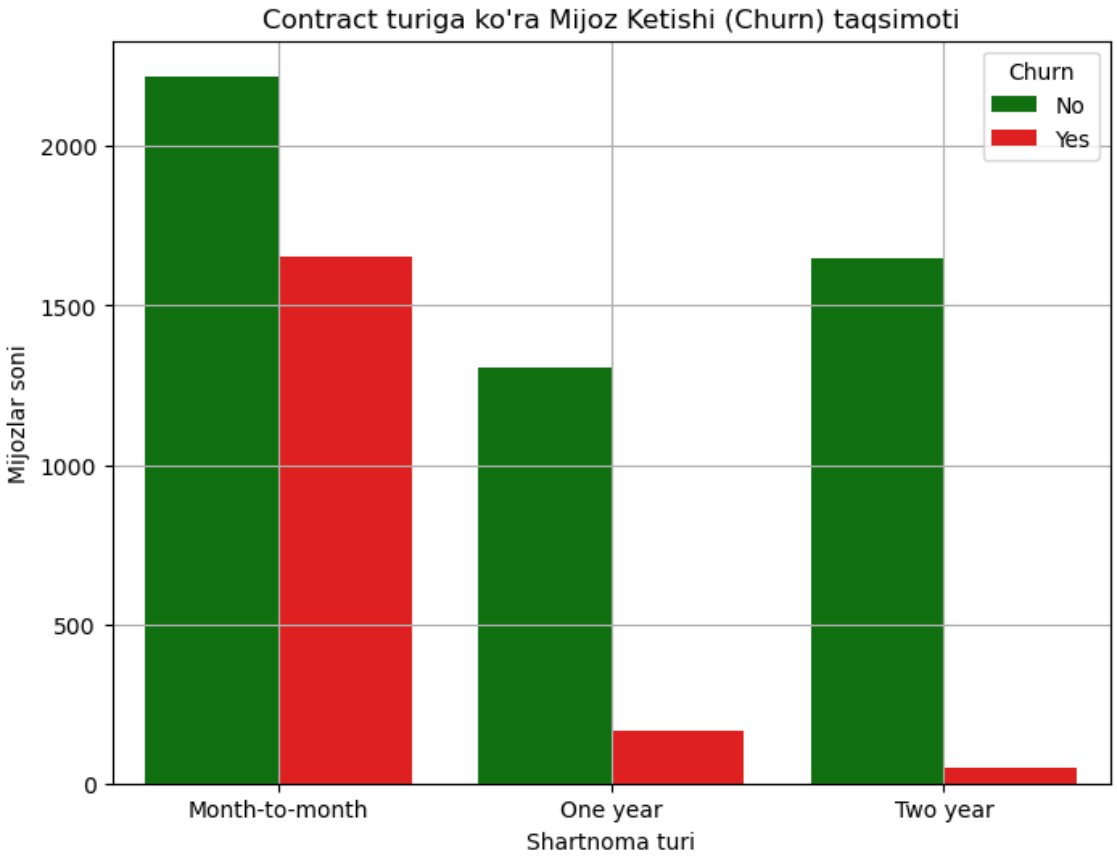
1. Tenure (kompaniyada qolish muddati) bo‘yicha Churn taqsimoti bo’yicha grafik.



2. MonthlyCharges va Churn o‘rtasidagi bog‘liqlik (boxplot)



2. Contract turiga ko‘ra Churn taqsimoti (countplot)



Kiyingi qadamga o’tamiz:

4. Ma’lumotlarni tozalash:

1.Yetishmayotgan qiymatlar (NaN)

```
# NaN qiymatlar sonini ko'rish
print("NaN qiymatlar soni:\n", df.isna().sum())

# NaN qiymatlar bor satrlarni olib tashlash
df_clean = df.dropna()
```

NaN qiymatlarni quyidagi berilgan kod orqali topib oldim va ularni tashlab yubordim.

2. Har bir ustun bo'yicha '??' yoki 'unknown' qiymatlar borligini tekshirish

```
# Har bir ustun bo'yicha '??' yoki 'unknown' qiymatlar borligini tekshirish
for col in df_clean.columns:
    print(f"{col} ustunidagi noto'g'ri qiymatlar soni:",
          df_clean[col].isin(['??', 'unknown']).sum())

# Barcha noto'g'ri qiymatli satrlarni chiqarib tashlash
df_clean = df_clean[~df_clean.isin(['??', 'unknown']).any(axis=1)]

```

Buni ham quyidagi berilgan kod orqali bajarib chiqdim.

3. Noodatiy yoki salbiy qiymatlar (TotalCharges > 10000, tenure < 0)

```
# TotalCharges ustuni > 10000 yoki tenure < 0 bo'lgan satrlar
df_clean = df_clean[(pd.to_numeric(df_clean['TotalCharges'], errors='coerce') <= 10000) &
                    (df_clean['tenure'] >= 0)]

```

```
# TotalCharges ustunini floatga o‘tkazish (ko‘pincha string bo‘ladi)
df_clean['TotalCharges'] = pd.to_numeric(df_clean['TotalCharges'], errors='coerce')

# Yana NaN bo'lganlar chiqsa, ularni olib tashlash
df_clean = df_clean.dropna()

# Tekshirish: ma'lumot turlari
print(df_clean.dtypes)
```

Bu bosqichda NaN qiymatlarni va keraksiz ustunlarni uchrib yuqotib oldim va kiyingi bosqichga utdim ya’ni bu bosqich xususiyatlar bilan ishlash bosqich deb atalar ekan.

5. Xususiyatlar bilan ishlash:

1. Kategorik ustunlarni kodlash (One-Hot yoki Label Encoding)

```
# Kategorik ustunlar (object yoki bool)
categorical_cols = df_clean.select_dtypes(include=['object', 'bool']).columns.tolist()
print("Kategorik ustunlar:", categorical_cols)
```

Birinchi bulib kategorik ustunlarni topib oldim va kiyin,

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df_encoded = df_clean.copy()

# faqat ikkilik ustunlarga label encoding qo'llash
for col in categorical_cols:
    if df_encoded[col].nunique() == 2:
        df_encoded[col] = le.fit_transform(df_encoded[col])



df_encoded = pd.get_dummies(df_encoded, columns=[col for col in categorical_cols if df_clean[col].nunique() > 2], drop_first=True)
```

Ularni **LabelEncoder** yordamida kodlab oldim.

2. Sonli ustunlarni masshtablash (Scaler orqali)

```
from sklearn.preprocessing import StandardScaler

# Sonli ustunlar (int yoki float)
numeric_cols = df_encoded.select_dtypes(include=['int64', 'float64']).columns.tolist()

# Xohlasangiz, 'Churn' ustunini olib tashlang (agar u target bo‘lsa)
if 'Churn' in numeric_cols:
    numeric_cols.remove('Churn')

# Masshtablash
scaler = StandardScaler()
df_encoded[numeric_cols] = scaler.fit_transform(df_encoded[numeric_cols])

```

6. Model yaratish:

Model yaratishda RandomForest va Logistic Regrision modellarini tanlab oldim va ulardan kelib chiqib uzimga kerakli bulgan modelni ya’ni yuqori natija berganini model sifatida saqlab oldim.

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score
```

Bu yerda ikkala model uchun kerakli bulgan kutubxonalarni importini ko’rishimiz mumkin.

```python
# 1. Logistic Regression
log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train, y_train)
log_preds = log_model.predict(X_test)
log_proba = log_model.predict_proba(X_test)[:,1]

# 2. Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
rf_proba = rf_model.predict_proba(X_test)[:,1]

# 3. Har bir model uchun metrikalarni hisoblash
log_scores = {
    'accuracy': accuracy_score(y_test, log_preds),
    'f1': f1_score(y_test, log_preds),
    'roc_auc': roc_auc_score(y_test, log_proba)
}

rf_scores = {
    'accuracy': accuracy_score(y_test, rf_preds),
    'f1': f1_score(y_test, rf_preds),
    'roc_auc': roc_auc_score(y_test, rf_proba)
}
```

```python
# Natijalarni chiroyli ko'rsatish
print("Model baholash natijalari:\n")

print("Logistic Regression:")
for metric, score in log_scores.items():
    print(f"  {metric.capitalize()}: {score:.4f}")

print("\nRandom Forest:")
for metric, score in rf_scores.items():
    print(f"  {metric.capitalize()}: {score:.4f}")

# Eng yaxshi modelni aniqlash (F1 asosida, istasangiz ROC-AUC asosida ham tanlashingiz mumkin)
best_by = 'f1'  # yoki 'roc_auc' yoki 'accuracy'

if rf_scores[best_by] > log_scores[best_by]:
    best_model = rf_model
    best_model_name = "Random Forest"
else:
    best_model = log_model
    best_model_name = "Logistic Regression"

print(f"\n✅ Eng yaxshi model: {best_model_name} ( {best_by.upper()} = {max(rf_scores[best_by], log_scores[best_by]):.4f} )")

```

```
Model baholash natijalari:

Logistic Regression:
  Accuracy: 0.7958
  F1: 0.5639
  Roc_auc: 0.8451

Random Forest:
  Accuracy: 0.7940
  F1: 0.5105
  Roc_auc: 0.8467

✅ Eng yaxshi model: Logistic Regression ( F1 = 0.5639 )
```

Tepadagi rasmlardan kurishingiz mumkin bu yerda eng yaxshi model bu Logistic Regression va ushani uchun shu modelni saqlab olaman.

```python
import joblib

# Eng yuqori F1 score asosida model tanlash (yoki roc_auc/accuracy asosida tanlashingiz mumkin)
if rf_scores['f1'] > log_scores['f1']:
    best_model = rf_model
    print("Random Forest eng yaxshi model sifatida tanlandi.")
else:
    best_model = log_model
    print("Logistic Regression eng yaxshi model sifatida tanlandi.")

# Modelni saqlash
import os
os.makedirs("models", exist_ok=True)
model_path = "models/best_churn_model.pkl"
joblib.dump(best_model, model_path)

print(f"Model saqlandi: {model_path}")

```

```
Logistic Regression eng yaxshi model sifatida tanlandi.
Model saqlandi: models/best_churn_model.pkl
```

7. Natijalarni tahlil qilish

```python
import pandas as pd

feature_importances = pd.Series(rf_model.feature_importances_, index=X_train.columns)
feature_importances = feature_importances.sort_values(ascending=False)
print(feature_importances.head(10))

```

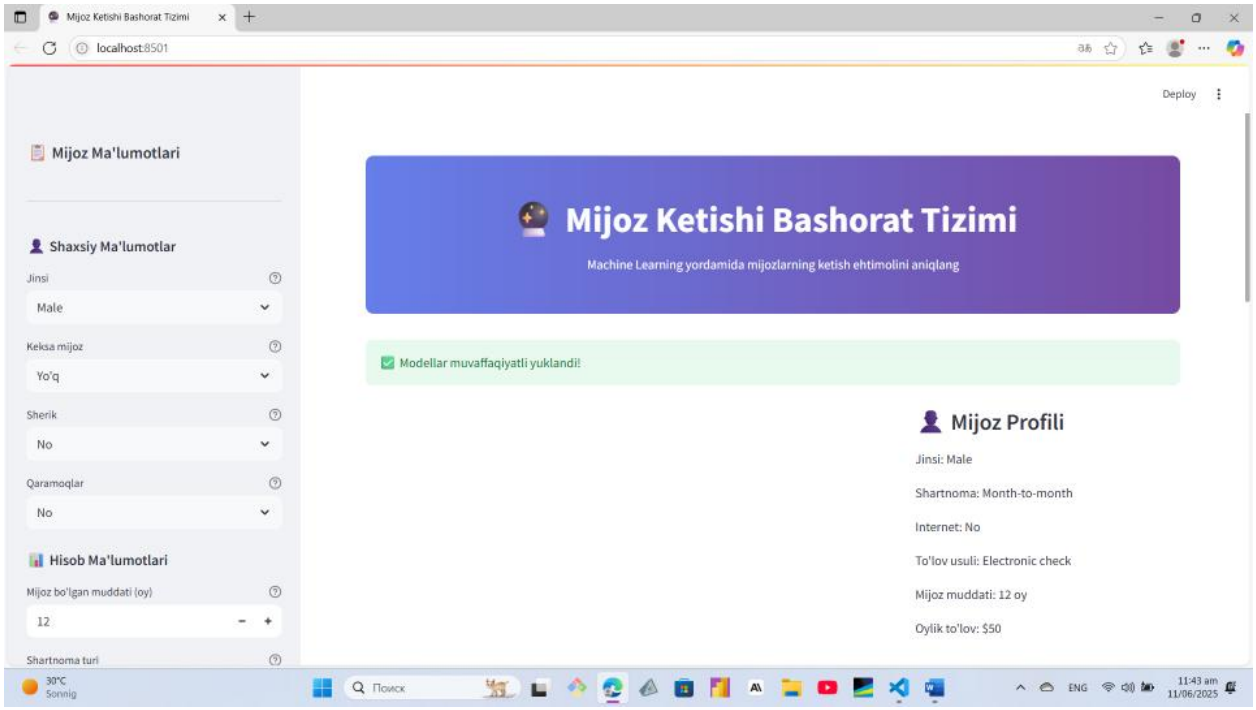
```
tenure                            0.082426
TotalCharges                      0.078735
InternetService_Fiber optic       0.027083
PaymentMethod_Electronic check    0.021987
OnlineSecurity_Yes                0.019495
Contract_Two year                 0.019237
TechSupport_Yes                   0.018823
Contract_One year                 0.016921
PaperlessBilling                  0.014669
Partner                           0.012633
dtype: float64
```

```python
# 3. Gipoteza: Contract turi va churn o'rtasidagi bog'liqlik
contract_churn_ct = pd.crosstab(df['Contract'], df['Churn'])

# Chi-kvadrat testi bilan bog'liqlikni tekshirish
chi2, p, dof, ex = chi2_contingency(contract_churn_ct)
print("Contract va Churn uchun Chi-kvadrat testi natijasi:")
print(f"Chi2={chi2}, p-value={p}")
```

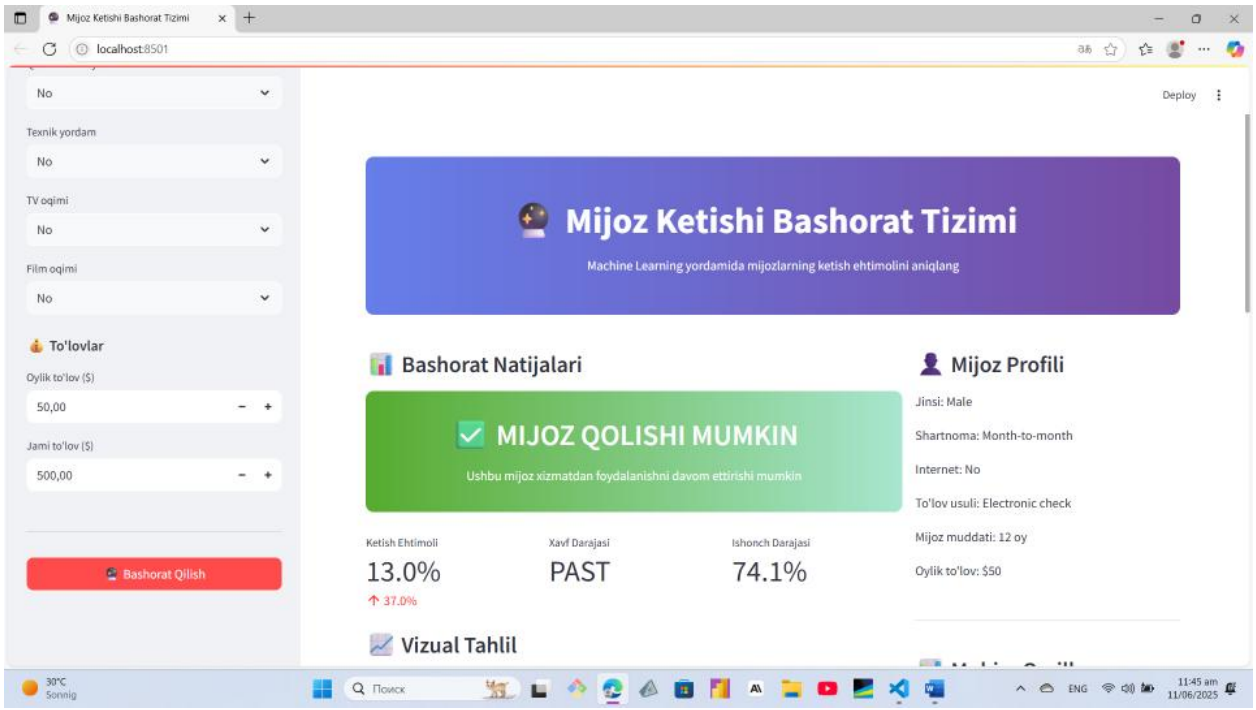
```
Contract va Churn uchun Chi-kvadrat testi natijasi:
Chi2=1184.5965720837926, p-value=5.863038300673391e-258
```

8. Oddiy tizim yaratish:

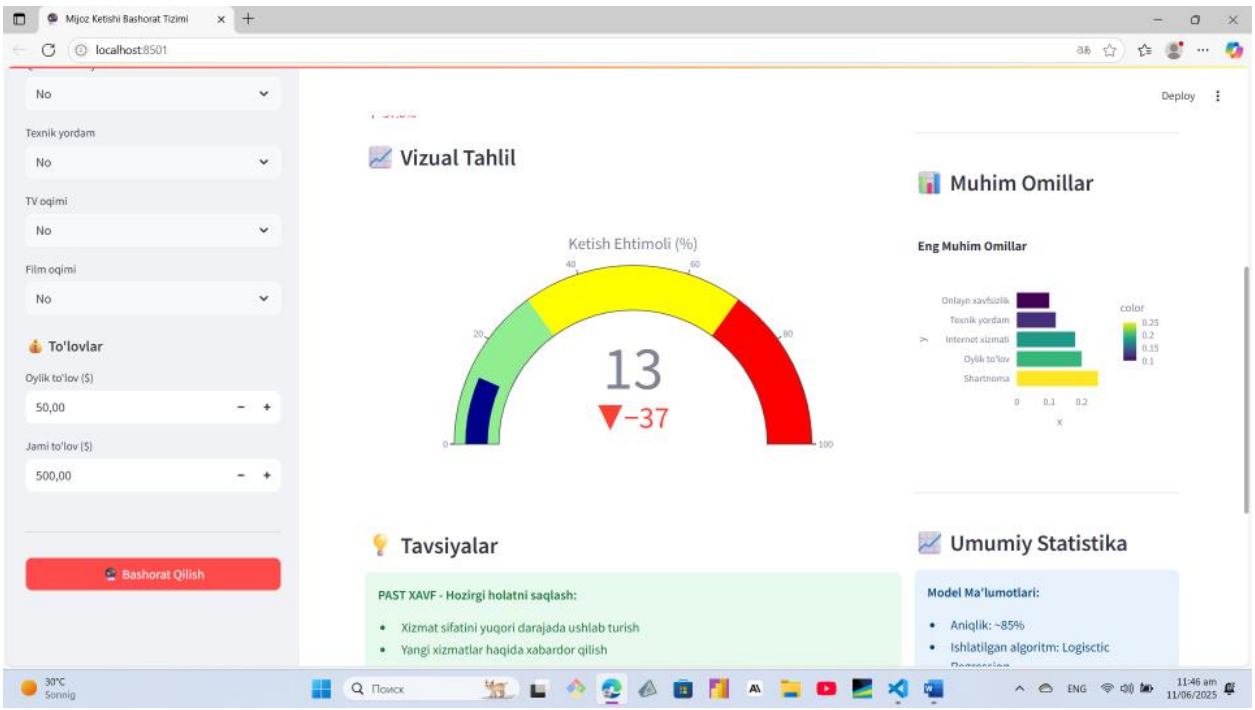
Birinchi bulib web-site yaratdim web-siteni ko’rinishi quydagicha:



Agar bashorat qiladigan bulsak quyidagi ko’rinishni beradi.

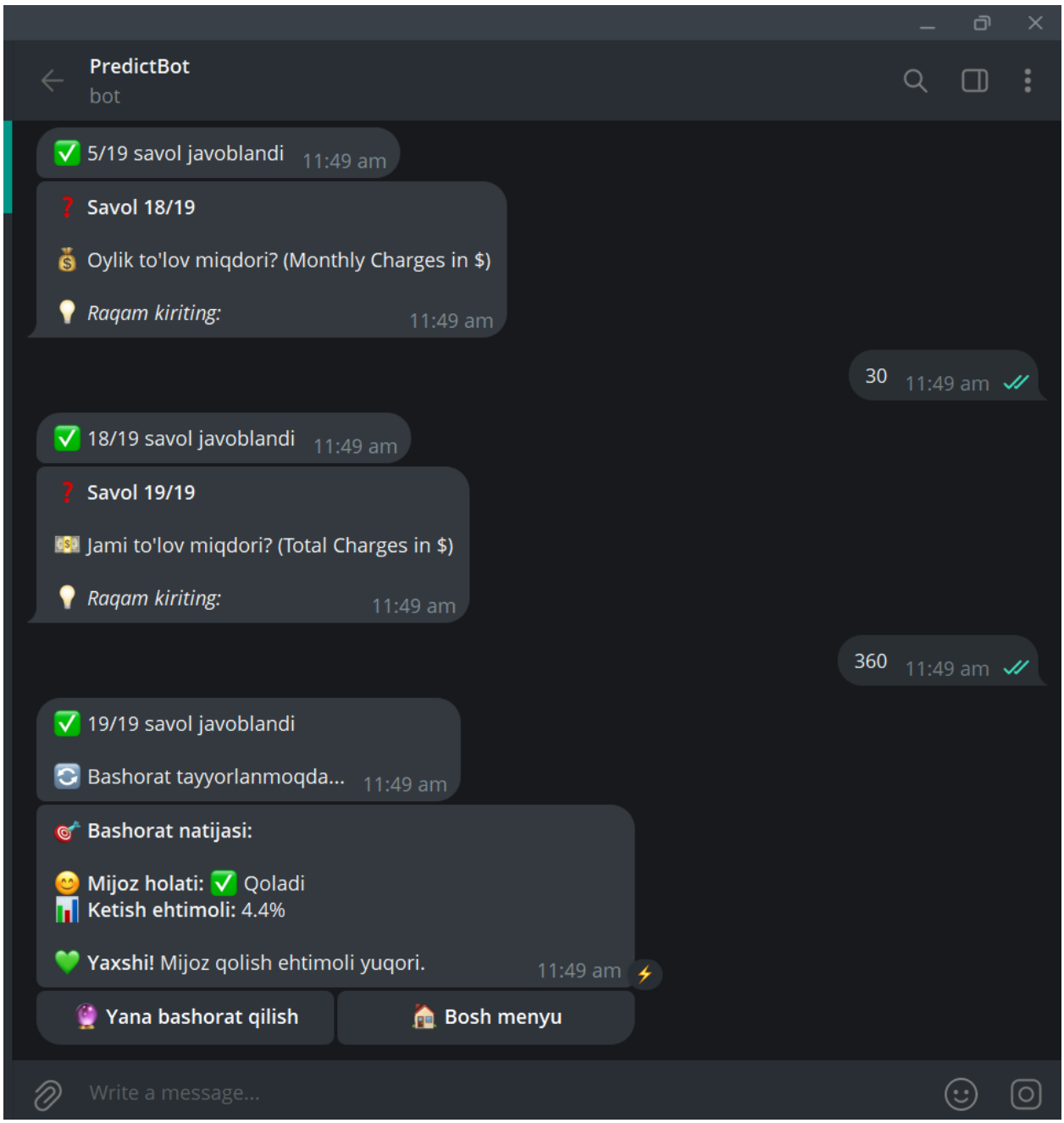


Bunda visual tahlil funksiyasi ham bor unda mijozning qolishi ketishi ehtimoligi kursatilgan.



2.Alohida telegram uchun ham bot qildim botda ham web-sitedagidaqa ishlaydi

Endi botni ishlashini kuramiz:


PredictBot
bot
✅ 5/19 savol javoblandi 11:49 am
❓ Savol 18/19
💰 Oylik to'lov miqdori? (Monthly Charges in $)
💡 Raqam kiriting: 11:49 am
30 11:49 am
✅ 18/19 savol javoblandi 11:49 am
❓ Savol 19/19
💵 Jami to'lov miqdori? (Total Charges in $)
💡 Raqam kiriting: 11:49 am
360 11:49 am
✅ 19/19 savol javoblandi
🔄 Bashorat tayyorlanmoqda... 11:49 am
🎯 Bashorat natijasi:
😊 Mijoz holati: ✅ Qoladi
📊 Ketish ehtimoli: 4.4%
💚 Yaxshi! Mijoz qolish ehtimoli yuqori. 11:49 am
🔮 Yana bashorat qilish
🏠 Bosh menyu
Write a message...