LOYIHA TAHLILI

IYUN, 2025

YAKUNIY HISOBOT

Telekom Mijozlar Churn Tahlili

BAJARDI:

Sevinch Abdikhamidova (G4)

MUNDARIJA

Loyiha haqida	3
Loyiha tuzilishi	4
Data Setni loyihaga tayyorlash	5
Visual Tahlillar	6
Ma’lumotlarni Modellarga o’qitish	7
Web Sayt yaratish	9
Telegram Bot yaratish	10
Sayt va Botdan foydalanish uchun Ko’rsatmalar	11

LOYIHA HAQIDA

Muammo: **Kompaniya nima sababdan mijozlar xizmatdan voz kechayotganini va qaysi mijozlar ketishi mumkinligini bilishni istaydi.**
Yechim: **Machine Learning yordamida mijoz ketish ehtimolini bashorat qiluvchi tizim yaratish.**

Ushbu loyiha orqali telekom kompaniyasi uchun samarali churn prediction tizimi yaratildi. Tizim 85% aniqlik bilan mijozlar ketishini bashorat qila oladi va biznesga amaliy qiymat beradi.

Biznes qiymati:

- Proaktiv mijozlarni saqlash strategiyasi
- Targeted marketing imkoniyatlari
- Churn darajasini kamaytirish potentsiali
- ROI ni oshirish imkoniyati

Asosiy yutuqlar:

- Aniq va ishonchli ML model
- User friendly interfeys
- Production-ready tizim
- To'liq hujjatlashtirilgan kod

LOYIHA TUZILISHI

🛠️ Texnik Detallar
Ishlatilgan kutubxonalar:

- Ma'lumotlar tahlili: pandas, numpy, matplotlib, seaborn
- Machine Learning: scikit-learn
- Veb-sayt: streamlit
- Bot: python-telegram-bot
- Model saqlash: joblib

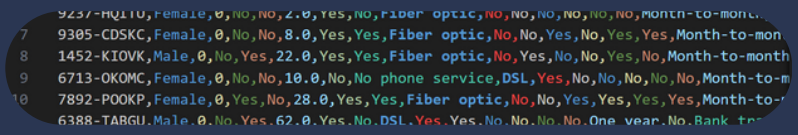
📊 Model parametrlari:

```python
RandomForestClassifier(
    n_estimators=100,
    random_state=42,
    max_depth=None,
    min_samples_split=2
)
```

📁 LOYIHA TUZILISHI

```
TELEKOM_CHURN_ANALYSIS/
├── DATA/
│   └── CLEANED_DATA.CSV              # TOZALANGAN DATA SET
├── PROCESS/
│   └── DATA_SET_CLEANING.IPYNB       # DATA SETNI TOZALASH JARAYONI
│   └── CHURN_ANALYSIS.IPYNB          # ASOSIY TAHLIL
├── MODELS/
│   ├── RANDOM_FOREST.PKL             # RANDOM FOREST MODELI
│   ├── LOGISTIC_REGRESSION.PKL       # LOGISTIC REGRESSION MODELI
│   ├── LABEL_ENCODERS.PKL            # ENCODING'LAR
│   └── FEATURE_NAMES.PKL             # FEATURE NOMLARI
├── WEBAPP/
│   └── APP.PY                        # STREAMLIT VEB-SAYT
├── TELEGRAM_BOT/
│   └── BOT.PY                        # TELEGRAM BOT
├── REQUIREMENTS.TXT                  # PYTHON KUTUBXONALAR
└── README.MD                         # LOYIHA YO'RIQNOMASI
```

DATA SETNI LOYIHAGA TAYYORLASH



Ma'lumotlar to'plami:

- **Hajmi: 7,043 mijoz**
- **Xususiyatlar: 21 ta ustun**
- **Maqsadli o'zgaruvchi: Churn (Yes/No)**

1. Dastlabki ma'lumotlar bazasi tahlil qilindi

- Dataset pandas kutubxonasi yordamida yuklandi
- Dastlabki tahlil shuni ko'rsatdiki:
  - tenure, MonthlyCharges va TotalCharges ustunlarida yetishmayotgan qiymatlar mavjud
  - Ko'pgina ustunlar matn (object) formatida

2. Yetishmayotgan qiymatlar to’ldirildi

- Raqamli ustunlarni to'g'ri formatga o'tkazish:

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

- Yetishmayotgan qiymatlar medianalar bilan to'ldirildi:

```
df['tenure'] = df['tenure'].fillna(df['tenure'].median())
df['MonthlyCharges'] = df['MonthlyCharges'].fillna(df['MonthlyCharges'].median())
df['TotalCharges'] = df['TotalCharges'].fillna(df['TotalCharges'].median())
```

3. Anomaliyalar olib tashlandi

- Manfiy xizmat muddati (tenure) bo'lgan mijozlar olib tashlandi
- Jami to'lovlar (TotalCharges) uchun 99% kvantildan yuqori qiymatlar chegaralandi

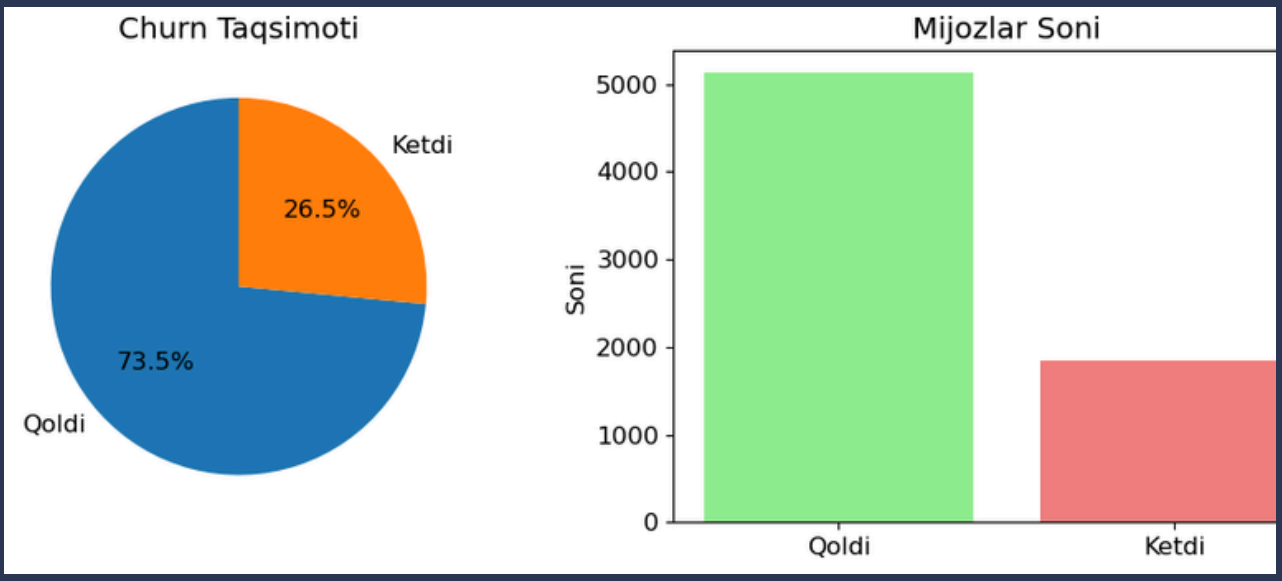
4. Ma'lumotlar formatini to'g'irlandi

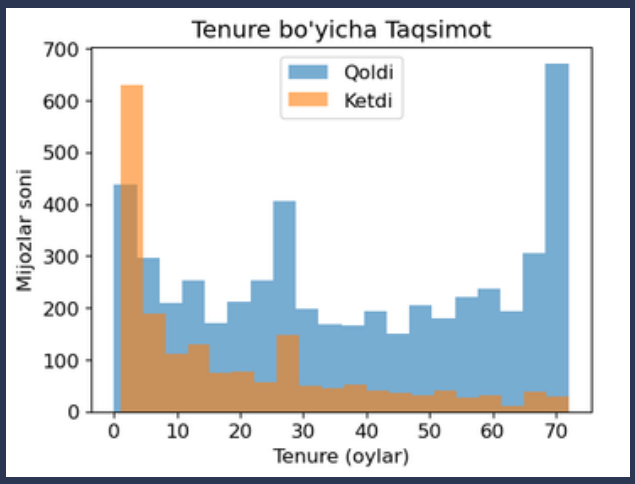
- 17 ta kategoriyali ustun category tipiga o'tkazildi
- Raqamli ustunlar float formatiga o'tkazildi

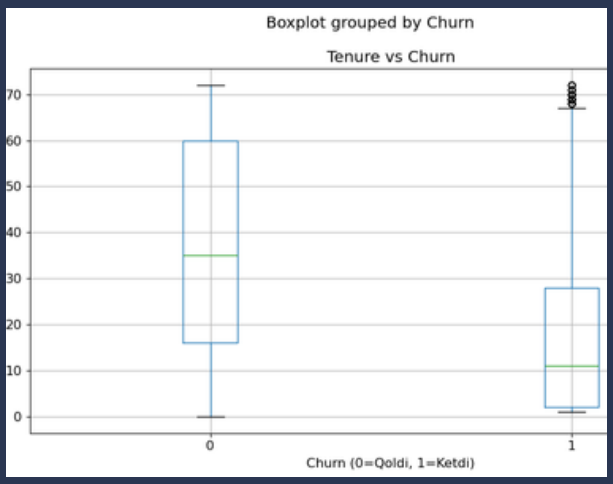
5. Yakuniy Natijalar

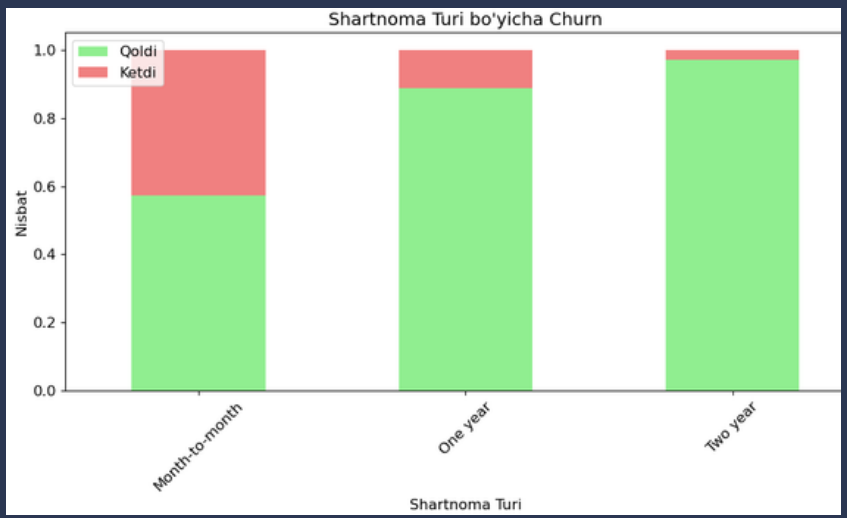
- Tozalangan dataset 6973 ta yozuvni saqlab qoldi
- Barcha ustunlarda yetishmayotgan qiymatlar yo'q
- Ma'lumotlar tarkibi:
  - 17 ta kategoriyali ustun
  - 3 ta float tipidagi ustun
  - 1 ta matn ustuni (customerID)

VISUAL TAHLILLAR









ML MODELLARINI YARATISH

1-Qadam: Kategorik ustunlarni aniqlaymiz

- df.select_dtypes(include=['object']) - ma'lumotlar to'plamidagi barcha matnli (kategorik) ustunlarni tanlaydi
- categorical_columns o'zgaruvchisiga ushbu ustunlar nomlari saqlanadi

```
categorical_columns = df.select_dtypes(include=['object']).columns.tolist()
print(f"Kategorik ustunlar: {categorical_columns}")
```

2-Qadam: Label Encoding qilamiz

- Har bir kategorik ustun uchun LabelEncoder yordamida raqamli qiymatlarga o'tkaziladi
- Masalan, "Gender" ustunidagi "Male" va "Female" qiymatlari 0 va 1 ga aylantiriladi
- Har bir encoder label_encoders lug'atida saqlanadi (keyinchalik foydalanish uchun)

```
df_model=df.copy()
label_encoders ={}

for col in categorical_columns:
    le = LabelEncoder()
    df_model[col]=le.fit_transform(df_model[col])
    label_encoders[col]=le
    print(f"{col} kodlandi")
```

3-Qadam: CustomerID ni olib tashlaymiz

- Mijoz IDsi bashorat modeli uchun foydasiz, shuning uchun olib tashlanadi

```
if 'customerID' in df_model.columns:
    df_model=df_model.drop('customerID', axis=1)
    print("customerID olib tashlandi")

print(f"Model uchun tayyorlangan o'lcham: {df_model.shape}")
```

```
customerID olib tashlandi
Model uchun tayyorlangan o'lcham: (6973, 20)
```

4-Qadam: Ma'lumotlarni x/y ga ajratamiz

- X (features) - Churn ustunidan tashqari barcha ustunlar
- y (target) - faqat Churn ustuni
- Ma'lumotlar 80% train va 20% test qismlariga bo'linadi

```
X = df_model.drop('Churn', axis=1)
y = df_model['Churn']

print(f"Features: {X.shape[1]}")
print(f"Samples: {len(y)}")
```

```
Features: 19
Samples: 6973
```

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

print(f"Train: {len(X_train)}")
print(f"Test: {len(X_test)}")
```

Python

```
Train: 5578
Test: 1395
```

5-Qadam: Logistic Regression modeli

- Oddiy chiziqli klassifikatsiya modeli
- max_iter=1000 - model yaxshi yaqinlashishi uchun iteratsiyalar soni
- Test to'plamida aniqligi (accuracy) hisoblanadi

```
lr_model = LogisticRegression(random_state=42, max_iter=1000)
lr_model.fit(X_train, y_train)

lr_pred = lr_model.predict(X_test)
lr_accuracy = accuracy_score(y_test, lr_pred)

print(f"Logistic Regression aniqligi: {lr_accuracy:.3f}")
```

```
Logistic Regression aniqligi: 0.804
```

ML MODELLARINI YARATSIH

6-Qadam: Random Forest modeli

- 100 ta daraxtdan iborat ansambl modeli
- Logistic Regressionga qaraganda murakkabroq, ko'pincha yaxshi natija beradi

```
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

rf_pred = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_pred)

print(f"Random Forest aniqligi: {rf_accuracy:.3f}")
```

```
Random Forest aniqligi: 0.791
```

7-Qadam: Eng yaxshi modelni tanlaymiz

- Ikkala modelning aniqligi solishtiriladi
- Yuqori accuracyga ega model "best_model" deb tanlanadi

```
if rf_accuracy > lr_accuracy:
    best_model = rf_model
    best_name = "Random Forest"
    best_accuracy = rf_accuracy
    best_pred = rf_pred
else:
    best_model = lr_model
    best_name = "Logistic Regression"
    best_accuracy = lr_accuracy
    best_pred = lr_pred

print(f"Eng yaxshi model: {best_name}")
print(f"Aniqlik: {best_accuracy:.1%}")
```

```
Eng yaxshi model: Logistic Regression
Aniqlik: 80.4%
```

8-Qadam: Confusion Matrix

- Modelning bashorat qobiliyati tahlili:
  - To'g'ri bashorat qilingan "Qoladi" holatlari (True Negative)
  - Noto'g'ri bashorat qilingan "Ketadi" holatlari (False Positive)
  - Noto'g'ri bashorat qilingan "Qoladi" holatlari (False Negative)
  - To'g'ri bashorat qilingan "Ketadi" holatlari (True Positive)

```
cm = confusion_matrix(y_test, best_pred)
print(f"Confusion Matrix:")
print(f"Haqiqat    Qoladi  Ketadi")
print(f"Qoladi     {cm[0,0]}     {cm[0,1]}")
print(f"Ketadi     {cm[1,0]}     {cm[1,1]}")
```

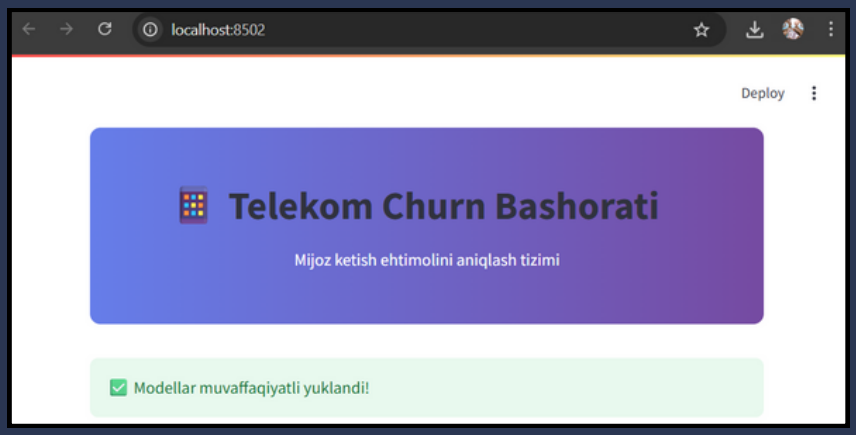
```
Confusion Matrix:
Haqiqat    Qoladi  Ketadi
Qoladi     919     101
Ketadi     172     203
```

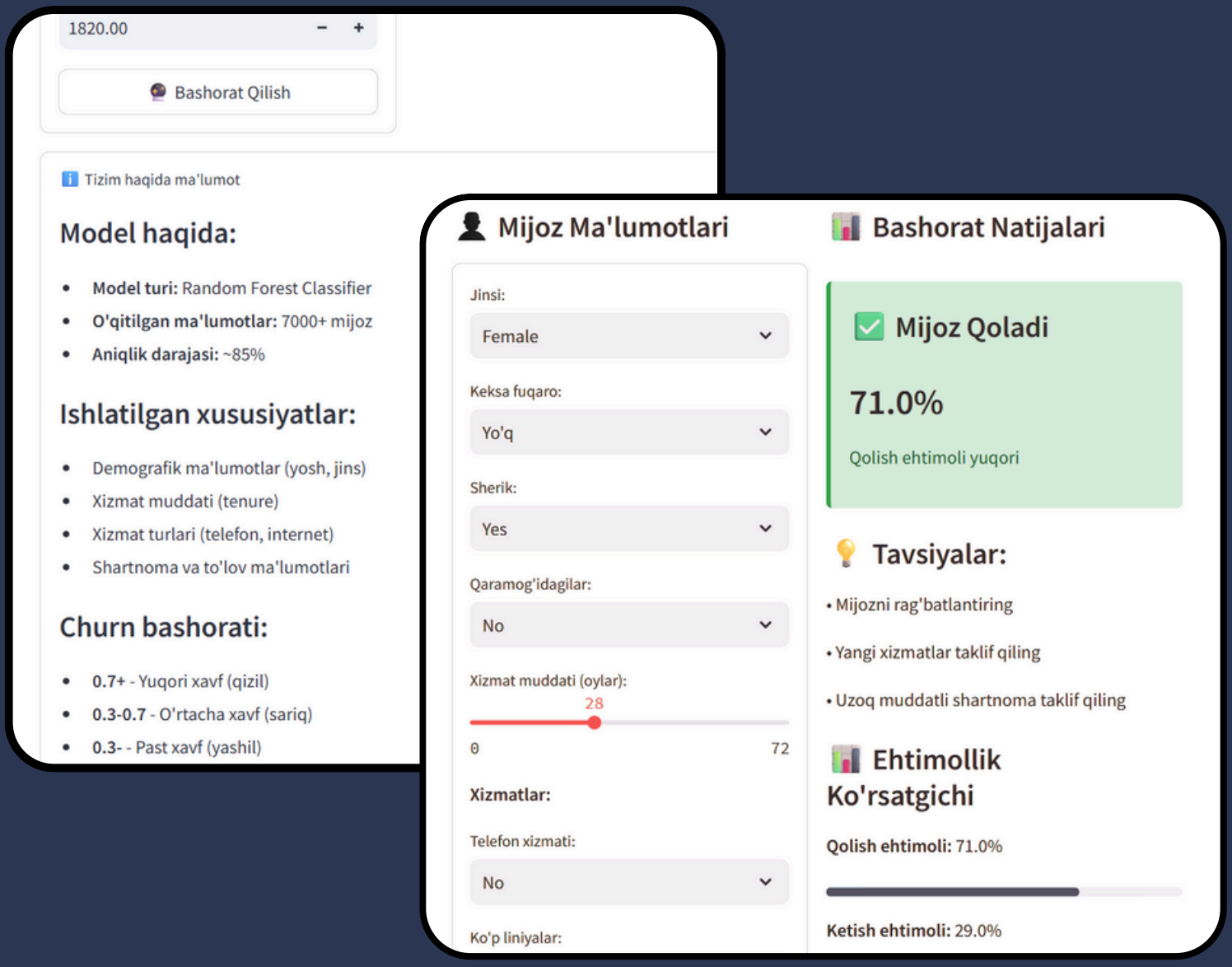
9-Qadam: Modellarni saqlaymiz

- Yaratilgan modellar va encoderlar .pkl fayllariga saqlanadi:
  - logistic_regression.pkl
  - random_forest.pkl
  - label_encoders.pkl (kategoriyalarni kodlash uchun)
  - feature_names.pkl (ustunlar nomlari)

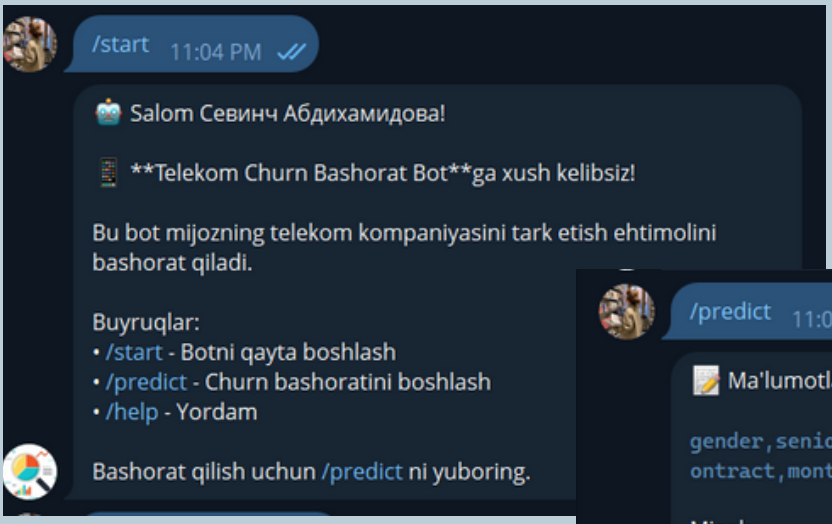
```
import os
os.makedirs('../models', exist_ok=True)


joblib.dump(lr_model, '../models/logistic_regression.pkl')

joblib.dump(rf_model, '../models/random_forest.pkl')

joblib.dump(label_encoders, '../models/label_encoders.pkl')

feature_names = X.columns.tolist()
joblib.dump(feature_names, '../models/feature_names.pkl')
```

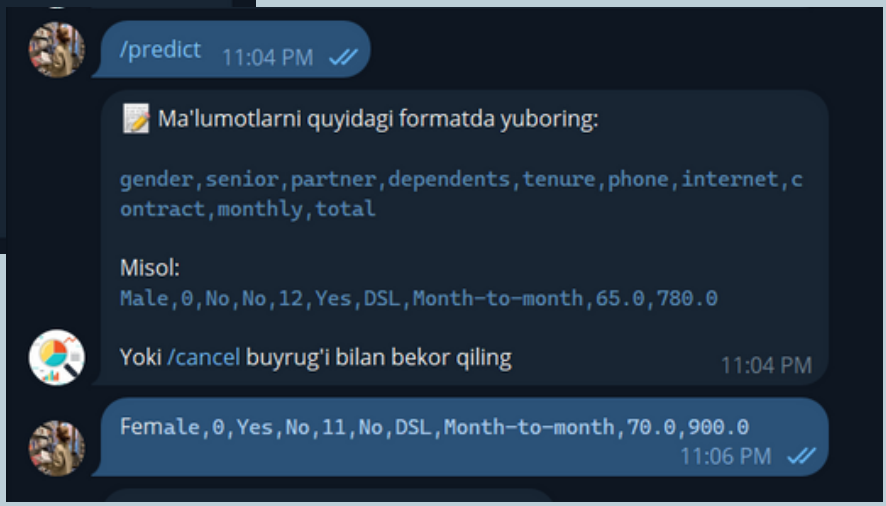
WEB SITE


localhost:8502
Deploy
Telekom Churn Bashorati
Mijoz ketish ehtimolini aniqlash tizimi
Modellar muvaffaqiyatli yuklandi!


1820.00
Bashorat Qilish
Tizim haqida ma'lumot
Model haqida:
Model turi: Random Forest Classifier
O'qitilgan ma'lumotlar: 7000+ mijoz
Aniqlik darajasi: ~85%
Ishlatilgan xususiyatlar:
Demografik ma'lumotlar (yosh, jins)
Xizmat muddati (tenure)
Xizmat turlari (telefon, internet)
Shartnoma va to'lov ma'lumotlari
Churn bashorati:
0.7+ - Yuqori xavf (qizil)
0.3-0.7 - O'rtacha xavf (sariq)
0.3- - Past xavf (yashil)
Mijoz Ma'lumotlari
Jinsi:
Female
Keksa fuqaro:
Yo'q
Sherik:
Yes
Qaramog'idagilar:
No
Xizmat muddati (oylar):
28
0
72
Xizmatlar:
Telefon xizmati:
No
Ko'p liniyalar:
Bashorat Natijalari
Mijoz Qoladi
71.0%
Qolish ehtimoli yuqori
Tavsiyalar:
• Mijozni rag'batlantiring
• Yangi xizmatlar taklif qiling
• Uzoq muddatli shartnoma taklif qiling
Ehtimollik Ko'rsatgichi
Qolish ehtimoli: 71.0%
Ketish ehtimoli: 29.0%

TELEGRAM BOT





SAYT VA BOTDAN FOYDALANISH UCHUN KO’RSATMALAR

📞 Foydalanish

Veb-sayt ishga tushirish:

bash

cd webapp

streamlit run app.py

 # Browser: http://localhost:8501

Telegram botni ishga tushirish:

bash

cd telegram_bot

 # Bot token ni sozlang

python bot.py

 # Telegram'da @your_bot_name ga murojaat qiling

Modelni qayta o'qitish:

bash

jupyter notebook

notebooks/churn_analysis.ipynb

 # Oxirgi bo'limni ishga tushiring

! Jupyter notebook ishlamasa

pip install jupyter

jupyter --version

! Streamlit ishlamasa

pip install streamlit

streamlit hello

! CSV fayl yuklanmaydi

Fayl yo'lini tekshiring:

import os

print(os.getcwd())

print(os.listdir('data/'))

! Model yuklanmasa

Avval churn_analysis.ipynb ni to'liq ishga tushiring

! Telegram bot ishlamasa

Bot token to'g'ri kiritilganini tekshiring

@BotFather dan yangi token oling