Eshimova Xurshidaning imtihon vazifasi hisoboti

1-vazifa. Biznes muammosini tushunish:
1.1. Nima uchun kompaniya uchun mijoz ketishini oldindan bilish muhim?
Mijoz ketishi - bu kompaniya xizmatidan mijozning butunlay voz kechishidir. Uni oldindan aniqlash juda muhim, chunki:

- Koʻp mijozlar ketadigan boʻlsa, kompaniyaning daromadi tushib ketadi.
- Oldindan xavfli mijozlarni aniqlab, ular bilan individual ish olib borish mumkin - chegirmalar, maxsus takliflar orqali.

1.2. Qaysi turdagi mijozlar "xavfli" toifaga kiradi?
Tahminan quyidagi mijozlar "xavfli" ya'ni ketish ehtimoli katta bo'lgan mijozlar deb qarash mumkin.

- Yangi roʻyxatdan oʻtgan, lekin kam foydalangan;
- Har oy oʻzi toʻlov qiladigan, avto-toʻlovdan foydalanmaydiganlar;
- Har oyda juda kam miqdorda yoki aksincha juda ko'p miqdorda to'lov qilayotgan mijozlar.

2-vazifa: Ma'lumotlarni tahlil qilish va gipotezalarni tekshirish.
2.1. Dastlabki ma'lumotlarda 21 ta ustun mavjud bo'lib, bo'sh qoldirilgan ma'lumotlar, anomaliyalar mavjud edi va datasetni tozalash lozim.

Umumiy oʻlcham: df.shape
Ustun nomlari va turlari: df.info()
Yetishmayotgan qiymatlar: df.isnull().sum()

2.2. Kamida 3 ta gipoteza taklif qiling va statistik usullar bilan tekshiring:

- **Masalan, yangi mijozlar koʻproq ketadimi?**

Nol gipoteza (H_0):
Ketgan mijozlar va qolgan mijozlarning xizmatdan foydalanish davomiyligi oʻrtacha bir xil (ya'ni yangi yoki eskiligi farq qilmaydi).
Alternativ gipoteza (H_1):
Ketgan mijozlar xizmatdan kamroq muddat foydalanishgan (ya'ni ular yangi mijozlar boʻlgan).
Bu gipotezani Student's t-test orqali tahlil qilamiz.

```python
import pandas as pd
from scipy import stats

ketgan = df[df['Churn'] == 'Yes']['tenure']
qolgan = df[df['Churn'] == 'No']['tenure']

t_stat, p_value = stats.ttest_ind(ketgan, qolgan, equal_var=False)

print("T-statistic:", t_stat)
print("P-value:", p_value)
```

✓ 0.0s

```
T-statistic: -32.682351268198886
P-value: 4.374068567142151e-209
```

P-value = 4.37e-209 bu degani:

- Bu qiymat **0.05 dan ancha kichik** (hatto ≈ 0 ga teng);
- Ya’ni, **nol gipotezani (H_0) qat’iyan rad qilamiz.**

T-statistic: -32.68

- **Salbiy chiqdi —** bu shuni anglatadiki, ketgan mijozlarning o‘rtacha “tenure” (xizmat davomiyligi) qolganlarnikidan ancha kichik.
- Ya’ni ular ko‘proq yangi mijozlar.

Demak yangi mijozlar ko’proq tark etishmoqda.

- **Internet xizmatidan foydalanuvchilar ko‘proq ketadimi?**

  Gipotezalarni aniqlash:

- H_0 (nol gipoteza): Internet xizmat turi va mijoz ketishi o‘rtasida bog‘liqlik yo‘q
- H_1 (muqobil gipoteza): Internet xizmat turi va mijoz ketishi o‘rtasida bog‘liqlik bor

InternetService” (internet turi) bilan “Churn” (xizmatdan ketish) o‘rtasidagi bog‘liqlik Chi-kvadrat testi yordamida tahlil qilindi.

Natijalarga ko‘ra, internet turi mijoz ketishiga sezilarli ta’sir ko‘rsatadi. Ayniqsa, Fiber optic xizmatidan foydalanuvchilar orasida xizmatdan voz kechish holatlari ko‘proq kuzatildi.

```python
import pandas as pd
from scipy.stats import chi2_contingency

df = df[['InternetService', 'Churn']].dropna()
table = pd.crosstab(df['InternetService'], df['Churn'])
# Chi-kvadrat testi
chi2, p, dof, expected = chi2_contingency(table)
print("Chi-squared:", chi2)
print("P-value:", p)
print("Degrees of Freedom:", dof)
print("\nContingency Table:\n", table)

```

✓ 0.3s

```
Chi-squared: 732.309589667794
P-value: 9.571788222840544e-160
Degrees of Freedom: 2

Contingency Table:
 Churn              No   Yes
InternetService
DSL              1962   459
Fiber optic      1799  1297
No               1413   113
```

- **Ayollar kamroq ketadimi?**

Gipotezalarni aniqlash:

- **H_0** (nol gipoteza): Mijozning jinsi (`gender`) va xizmatdan ketishi (`Churn`) o‘rtasida bog‘liqlik yo‘q
- **H_1** (muqobil gipoteza): Mijozning jinsi va xizmatdan ketishi o‘rtasida bog‘liqlik bor

```python
import pandas as pd
from scipy.stats import chi2_contingency
df_gender_churn = df[['gender', 'Churn']].dropna()

table = pd.crosstab(df_gender_churn['gender'], df_gender_churn['Churn'])
chi2, p, dof, expected = chi2_contingency(table)
print("Chi-squared:", chi2)
print("P-value:", p)
print("Degrees of Freedom:", dof)
print("\nContingency Table:\n", table)

```

✓ 0.0s

```
Chi-squared: 0.4840828822091383
P-value: 0.48657873605618596
Degrees of Freedom: 1

Contingency Table:
 Churn      No  Yes
gender
Female   2549  939
Male     2625  930
```

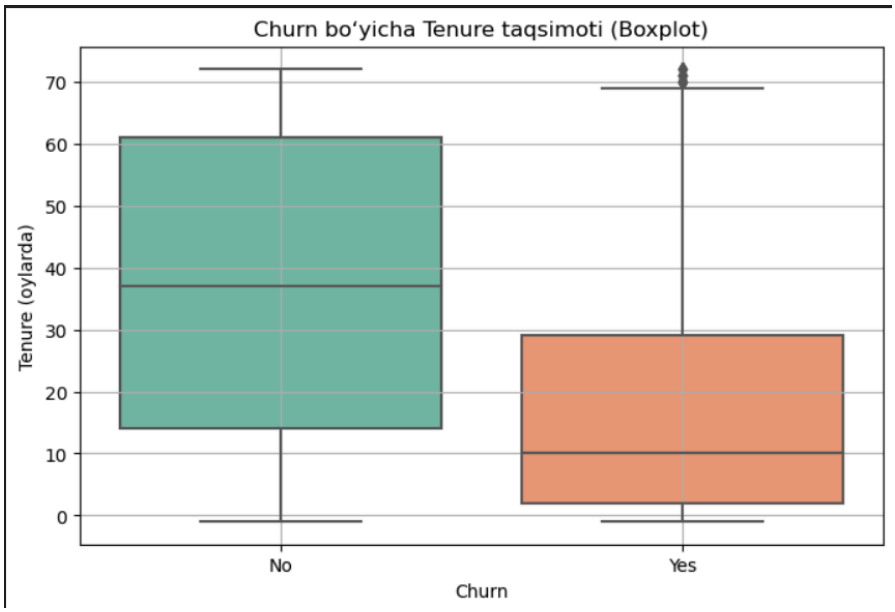
Chi-kvadrat testi natijalariga ko‘ra, mijozning jinsi (gender) bilan xizmatdan voz kechishi (Churn) o‘rtasida statistik ahamiyatga ega bog‘liqlik aniqlanmadi ($\chi^2 = 0.484$, df = 1, p = 0.487). Bu esa ayol va erkak mijozlar orasida ketish ehtimoli deyarli bir xil ekanini ko‘rsatadi.

3-vazifa. Vizualizatsiya:

3.1. tenure (kompaniyada qolish muddati) bo‘yicha Churn taqsimoti

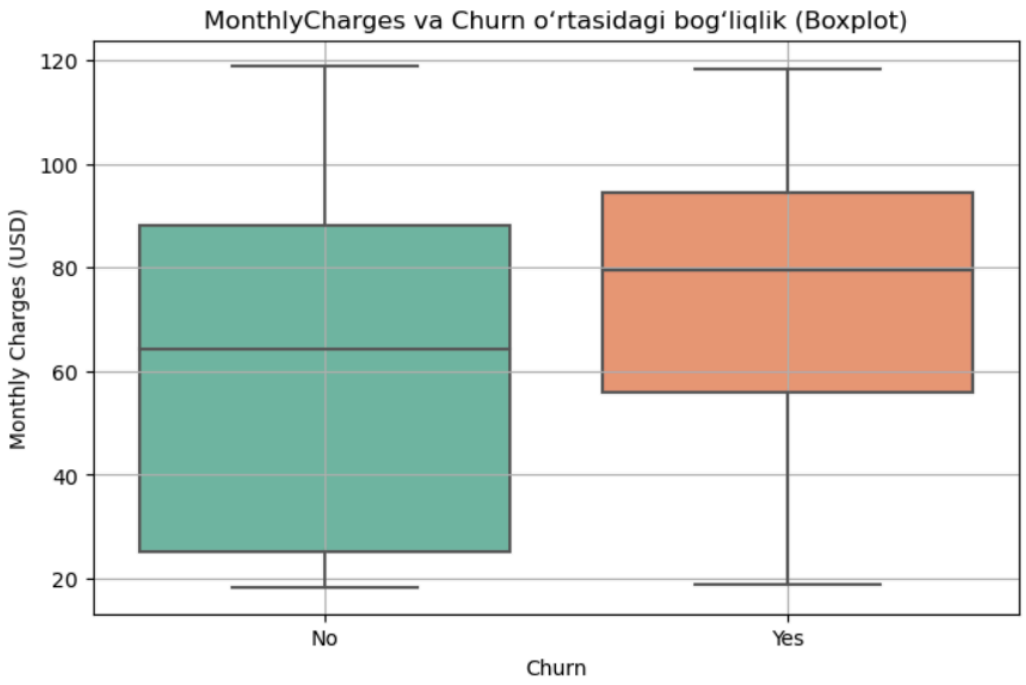
```python
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8, 5))
sns.boxplot(x='Churn', y='tenure', data=df,
palette='Set2')
```

```
plt.title('Churn bo‘yicha Tenure taqsimoti (Boxplot)')
plt.xlabel('Churn')
plt.ylabel('Tenure (oylarda)')
plt.grid(True)
plt.show()
```



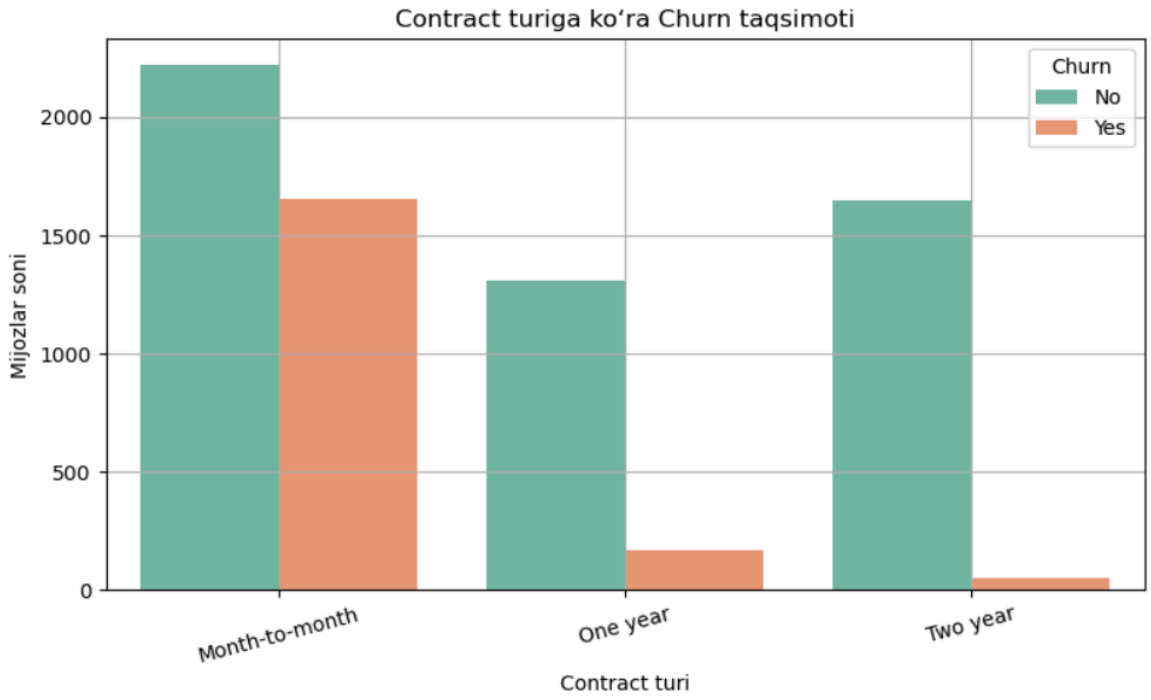
3.2. MonthlyCharges va Churn o‘rtasidagi bog‘liqlik (boxplot)

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.boxplot(x='Churn', y='MonthlyCharges', data=df, palette='Set2')
plt.title('MonthlyCharges va Churn o‘rtasidagi bog‘liqlik (Boxplot)')
plt.xlabel('Churn')
plt.ylabel('Monthly Charges (USD)')
plt.grid(True)
plt.show()
```



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

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='Contract', hue='Churn', palette='Set2')
plt.title('Contract turiga ko‘ra Churn taqsimoti')
plt.xlabel('Contract turi')
plt.ylabel('Mijozlar soni')
plt.legend(title='Churn')
plt.grid(True)
plt.xticks(rotation=15)
plt.tight_layout()
plt.show()
```



4. Ma’lumotlarni tozalash:

• Yetishmayotgan qiymatlar (NaN)

• Noto‘g‘ri qiymatlar (??, unknown)

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

• To‘g‘ri ma’lumot turlari bilan ishlash

```
import pandas as pd
df = pd.read_csv('data.csv')

df = df.drop(['customerID'], axis=1)
df = df.drop(['gender'], axis=1)



df['MonthlyCharges'] = df['MonthlyCharges'].replace('??', 'NaN')

df['MonthlyCharges'] = df['MonthlyCharges'].astype('float')

df['TotalCharges'] = df['TotalCharges'].replace('??', 'NaN')
df['TotalCharges'] = df['TotalCharges'].replace(' ', 'NaN')
```

```
df['TotalCharges'] = df['TotalCharges'].astype('float')

df['tenure'] = df['tenure'].replace(-1.0, 1.0)
df['tenure'] = df['tenure'].fillna(1.0)

fiber_mode = df[df['InternetService'] == 'Fiber
optic']['MonthlyCharges'].mean()
dsl_mode = df[df['InternetService'] == 'DSL']['MonthlyCharges'].mean()
no_mode = df[df['InternetService'] == 'No']['MonthlyCharges'].mean()

# 2. Shartli ravishda to‘ldiramiz
df.loc[(df['InternetService'] == 'Fiber optic') &
(df['MonthlyCharges'].isna()), 'MonthlyCharges'] = fiber_mode
df.loc[(df['InternetService'] == 'DSL') & (df['MonthlyCharges'].isna()),
'MonthlyCharges'] = dsl_mode
df.loc[(df['InternetService'] == 'No') & (df['MonthlyCharges'].isna()),
'MonthlyCharges'] = no_mode

# Avval TotalCharges ustunini tozalaymiz
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')

# Anormal yoki bo‘sh qiymatlar o‘rniga qayta hisoblaymiz
mask = (df['TotalCharges'].isna()) | (df['TotalCharges'] >
df['MonthlyCharges'] * df['tenure'] * 2)

df.loc[mask, 'TotalCharges'] = df.loc[mask, 'MonthlyCharges'] *
df.loc[mask, 'tenure']
```

1. Keraksiz ustunlarni olib tashlash

- **`customerID`** – har bir mijozga unikal identifikator bo‘lgani uchun model uchun foydali emas, olib tashlandi.

- **`gender`** – ushbu ustun asosida tahlil alohida bajarilgan bo‘lib, modelga ta’siri past deb baholanib, chiqarib yuborildi.

2. Noto‘g‘ri qiymatlarni aniqlash va tuzatish

- **`MonthlyCharges` va `TotalCharges`** ustunlarida ba’zi qiymatlar `'??'` yoki bo‘sh joy (`' '`) ko‘rinishida ifodalangan bo‘lib, ular **`NaN`** qiymatiga almashtirildi.

- Ushbu ustunlar **float** (haqiqiy son) formatiga o‘tkazildi.

3. `tenure` ustunidagi g‘ayritabiiy qiymatlar

- `tenure` (kompaniyada qolgan oylar soni) ustunida ayrim qiymatlar **-1.0** edi. Bu qiymat mavjud emas deb qabul qilinib, **1.0** ga almashtirildi.

- `NaN` bo‘lgan `tenure` qiymatlari ham **1.0** bilan to‘ldirildi.

4. MonthlyCharges ustunidagi `NaN` qiymatlarni to‘ldirish

- `MonthlyCharges` ustunidagi bo‘sh qiymatlar **InternetService** turiga qarab o‘rtacha qiymat bilan to‘ldirildi:

    - **Fiber optic** foydalanuvchilari uchun – `fiber_mode`

    - **DSL** foydalanuvchilari uchun – `dsl_mode`

    - **No internet** foydalanuvchilari uchun – `no_mode`

Bu metod shartli to‘ldirish (conditional imputation) usuli hisoblanadi va mavjud bog‘liqlikni saqlashga yordam beradi.

5. TotalCharges ustunidagi anomaliyalarni tuzatish

- `TotalCharges` ustunidagi noto‘g‘ri yoki bo‘sh qiymatlar aniqlanib:

- qiymat **yo‘q (NaN)** bo‘lsa yoki
- **TotalCharges > MonthlyCharges × tenure × 2** bo‘lsa (ya'ni o‘zgaruvchilarning fizik mantiqiga zid),
- ularning qiymati `TotalCharges = MonthlyCharges × tenure` formulasi asosida qayta hisoblab chiqarildi.

5. Xususiyatlar bilan ishlash:

• Kategorik ustunlarni kodlash (One-Hot yoki Label Encoding)

• Sonli ustunlarni masshtablash (Scaler orqali)

6. Model yaratish:

• Kamida 2 ta modelni sinab ko‘ring:

o Logistic Regression

o Random Forest yoki XGBoost

• Model sifatini quyidagi mezonlar bo‘yicha baholang:

o Accuracy, F1 score, ROC-AUC, confusion matrix

Model o’qitilgan kodi:

```
import pandas as pd
```

```
import dill

from sklearn.pipeline import Pipeline

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer
```

```
from sklearn.model_selection import cross_val_predict, StratifiedKFold

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, f1_score, roc_auc_score,
confusion_matrix

from xgboost import XGBClassifier



def main():

    df = pd.read_csv('C:/exam/new_dataset2.csv')



    x = df.drop('Churn', axis=1)

    y = df['Churn']



    le = LabelEncoder()

    y = le.fit_transform(y)



    numerical_features = x.select_dtypes(include=['int64',
'float64']).columns

    categorical_features = x.select_dtypes(include=['object']).columns



    numerical_transformer = Pipeline(steps=[('scaler', StandardScaler())])

    categorical_transformer = Pipeline(steps=[('onehot',
OneHotEncoder(handle_unknown='ignore'))])
```

```
preprocessor = ColumnTransformer(transformers=[
    ('numerical', numerical_transformer, numerical_features),
    ('categorical', categorical_transformer, categorical_features)
])

models = (
    LogisticRegression(solver='liblinear'),
    RandomForestClassifier(random_state=42),
    XGBClassifier(use_label_encoder=False, eval_metric='logloss')
)

best_score = 0.0
best_model_name = ""
best_pipe = None
results = []

cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

for model in models:
    pipe = Pipeline(steps=[
        ('preprocessor', preprocessor),
        ('classifier', model)
```

```
        ])


        # Cross-validation orqali bashoratlar (test foldlar bo‘yicha)
        y_pred = cross_val_predict(pipe, x, y, cv=cv)


        roc_auc = roc_auc_score(y, y_pred)
        acc = accuracy_score(y, y_pred)
        f1 = f1_score(y, y_pred)
        cm = confusion_matrix(y, y_pred)


        results.append({
            'model': type(model).__name__,
            'accuracy': acc,
            'f1': f1,
            'roc_auc': roc_auc,
            'confusion_matrix': cm
        })


        if f1 > best_score:
            best_score = f1
            best_model_name = type(model).__name__
            # To‘liq modelni butun datasetda fit qilamiz
            best_pipe = pipe.fit(x, y)
```

```
    for res in results:
        print(f"\nModel: {res['model']}")
        print(f"  Accuracy: {res['accuracy']:.4f}")
        print(f"  F1 Score: {res['f1']:.4f}")
        print(f"  ROC-AUC: {res['roc_auc']:.4f}")
        print(f"  Confusion Matrix:\n{res['confusion_matrix']}")

    # Eng yaxshi modelni faylga saqlash
    if best_pipe is not None:
        with open('best_model_pipeline.dill', 'wb') as f:
            dill.dump(best_pipe, f)
        print(f"\nEng yaxshi model '{best_model_name}' faylga saqlandi.")

if __name__ == '__main__':
    main()
```

Ushbu bosqichda mijozlarning xizmatdan voz kechishini (Churn) bashorat qiluvchi model qurildi va sinovdan o‘tkazildi. Jarayon quyidagi qadamlarni o‘z ichiga oladi:

1. Ma’lumotlarni yuklash va tayyorlash

- Ma’lumotlar `new_dataset2.csv` fayldan o‘qildi.

- Belgilangan maqsadli ustun — `Churn` (yo‘q yoki ha).
- Belgilangan maqsadli o‘zgaruvchi `LabelEncoder` yordamida raqamli formatga o‘tkazildi (`0` va `1`).

2. Xususiyatlarni turkumlash

- **Sonli ustunlar**: `int64`, `float64` tipidagi ustunlar.
- **Kategoriya ustunlari**: `object` tipidagi ustunlar.

3. Ma’lumotlarni oldindan qayta ishlash (preprocessing)

- **Sonli ustunlar** uchun `StandardScaler` ishlatilib, qiymatlar standartlashtirildi (o‘rtacha 0, dispersiya 1 bo‘ladi).
- **Kategoriya ustunlari** uchun `OneHotEncoder` qo‘llanilib, nomutanosib kategoriyalar uchun **ignore_unknown=True** parametri yordamida yangi kategoriyalar paydo bo‘lsa xatolik bo‘lmasligi ta’minlandi.
- Bu ikki bosqich `ColumnTransformer` yordamida birlashtirildi.

4. Modellarni tayyorlash va solishtirish

- Quyidagi uchta mashhur klassifikatsiya algoritmlari tanlandi:
    - **Logistic Regression** (`liblinear` solver bilan)
    - **Random Forest Classifier** (`random_state=42` bilan)

- **XGBoost Classifier** (`use_label_encoder=False` va `eval_metric='logloss'` parametrlari bilan)

5. Model baholash

- Har bir model uchun **5 ta stratifikatsiyalangan kross-valyadatsiya (StratifiedKFold)** bajarildi.

- Har bir qadamda quyidagi metrikalar hisoblandi:

1. **Accuracy** — to‘g‘ri klassifikatsiya ulushi.
2. **F1-Score** — aniqlik va chaqqonlikning uyg‘unligi, ayniqsa sinf nomutanosibligi holatlarida muhim.
3. **ROC-AUC** — modelning ajrata olish qobiliyatining o‘lchovi.
4. **Confusion Matrix** — haqiqiy va bashorat qilingan natijalar taqsimoti.

6. Eng yaxshi modelni tanlash va saqlash

- Eng yuqori **F1-score** ko‘rsatkichiga ega model tanlandi.

- Ushbu model butun ma’lumotlar to‘plamida o‘qitildi (`fit` qilindi).

- Natijada yaratilgan `Pipeline` (preprocessing + model) `best_model_pipeline.dill` fayliga saqlandi.

```
Model: LogisticRegression
  Accuracy: 0.8012
  F1 Score: 0.5935
  ROC-AUC: 0.7200
  Confusion Matrix:
[[4621  553]
 [ 847 1022]]

Model: RandomForestClassifier
  Accuracy: 0.7798
  F1 Score: 0.5391
  ROC-AUC: 0.6857
  Confusion Matrix:
[[4585  589]
 [ 962  907]]

Model: XGBClassifier
  Accuracy: 0.7758
  F1 Score: 0.5417
  ROC-AUC: 0.6875
  Confusion Matrix:
[[4531  643]
 [ 936  933]]

Eng yaxshi model 'LogisticRegression' faylga saqlandi.
```

8. Oddiy tizim yaratish:

Variant 1:

Veb-ilova :

• Mijoz ma’lumotlarini kiritish

• "Bashorat qilish" tugmasi

• Natija: ketadi / ketmaydi + ehtimol foizda

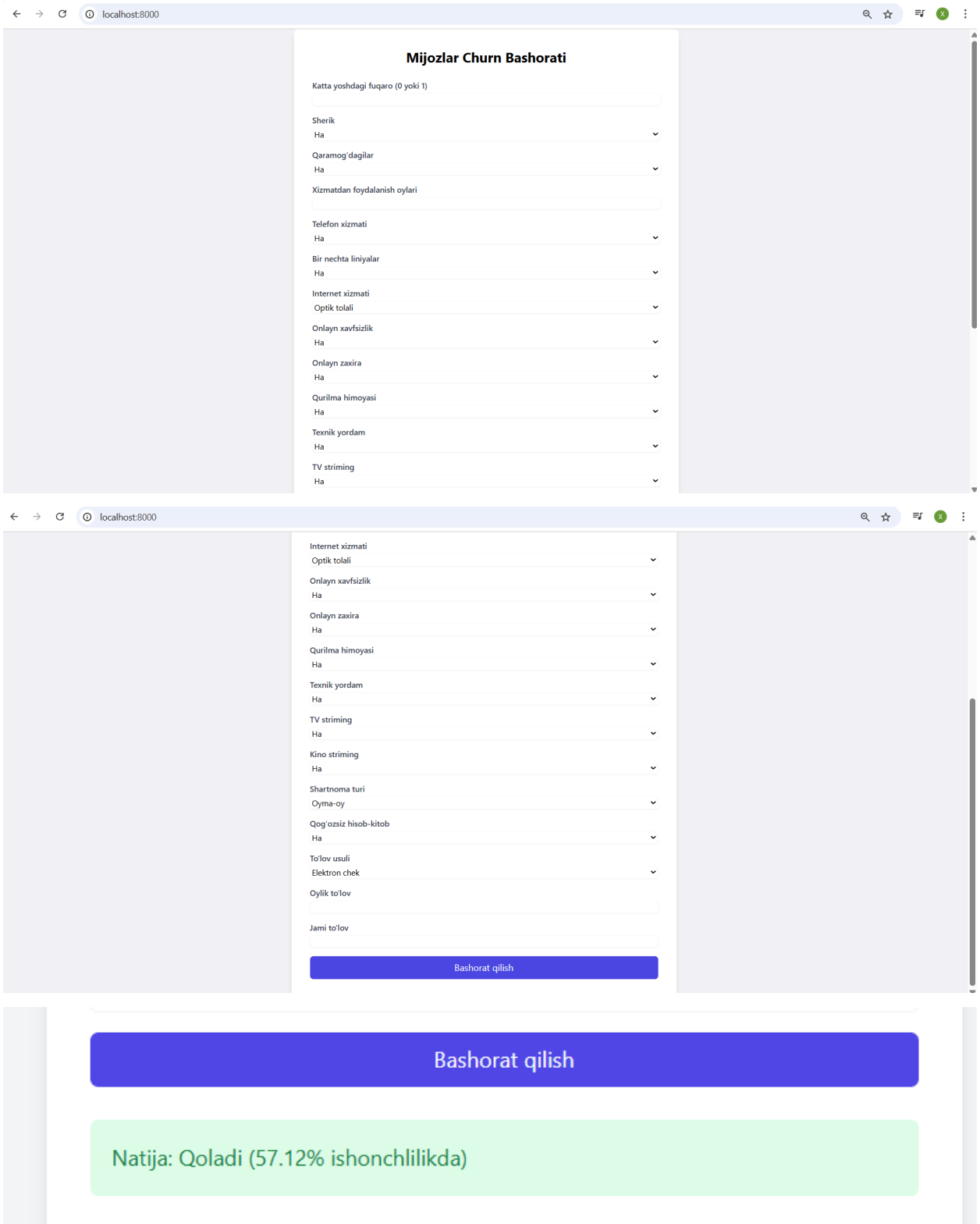
Variant 2:

Telegram-bot:

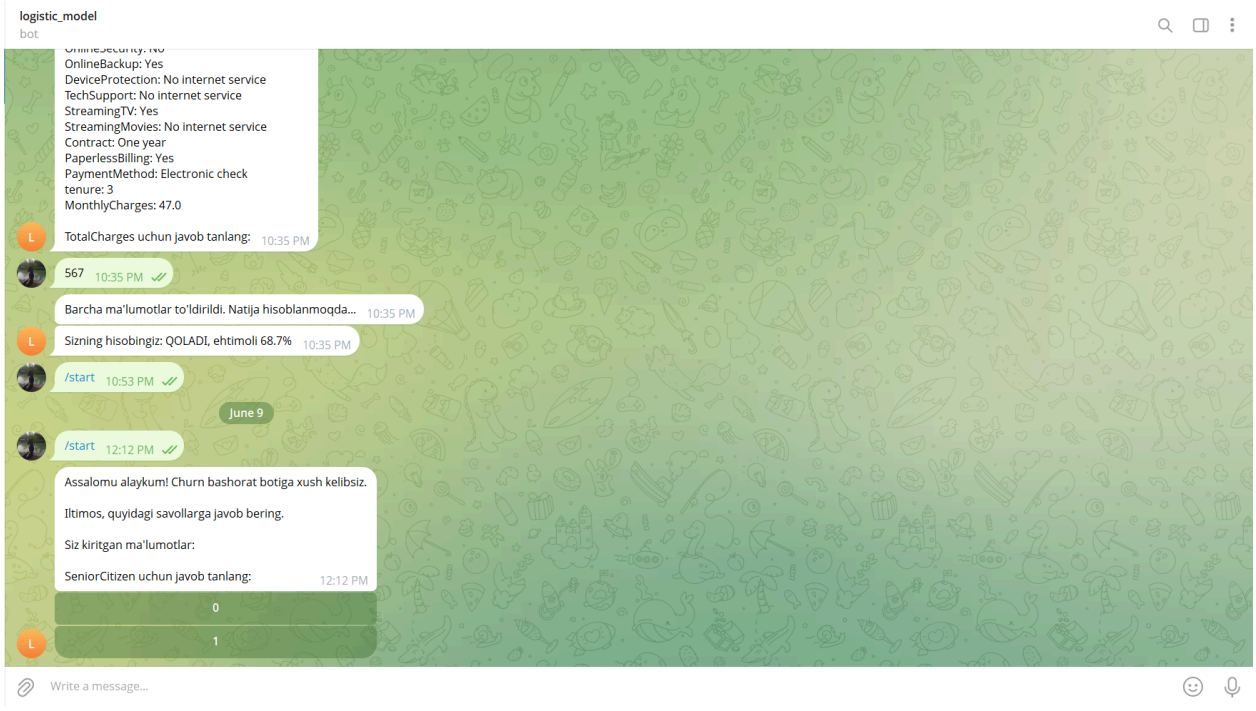
• /predict buyrug‘i orqali ma’lumotlar so‘raladi

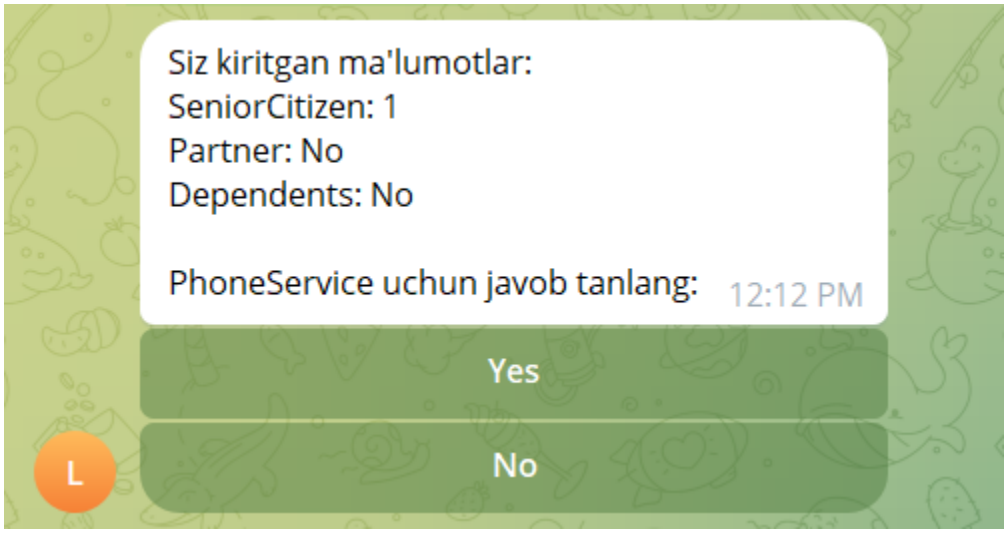
• Bot natijani qaytaradi

Web sayt skrenshoti:



Telegram bot skrenshoti:





logistic_model
bot

Partner: No
Dependents: No
PhoneService: Yes
MultipleLines: Yes
InternetService: Fiber optic
OnlineSecurity: No
OnlineBackup: No
DeviceProtection: No internet service
TechSupport: No
StreamingTV: No
StreamingMovies: No
Contract: Month-to-month
PaperlessBilling: No
PaymentMethod: Electronic check

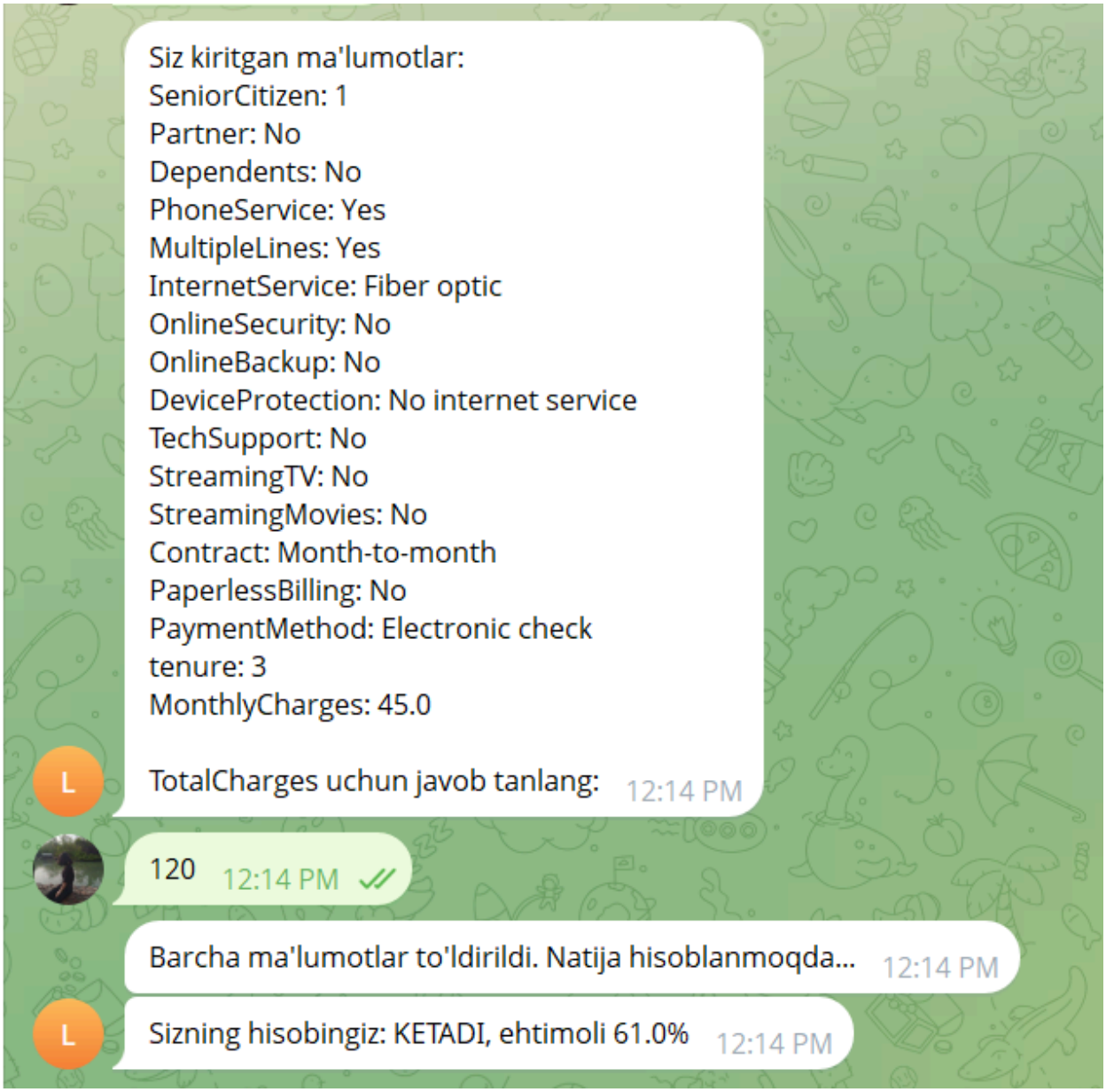
L tenure uchun javob tanlang: 12:12 PM

3 12:14 PM ✓✓

Siz kiritgan ma'lumotlar:
SeniorCitizen: 1
Partner: No
Dependents: No
PhoneService: Yes
MultipleLines: Yes
InternetService: Fiber optic
OnlineSecurity: No
OnlineBackup: No
DeviceProtection: No internet service
TechSupport: No
StreamingTV: No
StreamingMovies: No
Contract: Month-to-month
PaperlessBilling: No
PaymentMethod: Electronic check
tenure: 3

L MonthlyCharges uchun javob tanlang: 12:14 PM ❤

Write a message...


Siz kiritgan ma'lumotlar:
SeniorCitizen: 1
Partner: No
Dependents: No
PhoneService: Yes
MultipleLines: Yes
InternetService: Fiber optic
OnlineSecurity: No
OnlineBackup: No
DeviceProtection: No internet service
TechSupport: No
StreamingTV: No
StreamingMovies: No
Contract: Month-to-month
PaperlessBilling: No
PaymentMethod: Electronic check
tenure: 3
MonthlyCharges: 45.0
TotalCharges uchun javob tanlang: 12:14 PM
120 12:14 PM
Barcha ma'lumotlar to'ldirildi. Natija hisoblanmoqda... 12:14 PM
Sizning hisobingiz: KETADI, ehtimoli 61.0% 12:14 PM