```python
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# P—P°PiPsP»PsPIPsPє PїCЂPёP»PsP¶PµPSPёCЏ
st.title('P”PµPjPsPSCЃC,CЂP°C†PёCЏ PjPsPґPµP»PµP№ PjP°C€PёPSPSPsPiPs PsP±CґC‡PµPSPёCЏ')
st.write("""P-C,Ps PїCЂPёP»PsP¶PµPSPёPµ PїPsP·PIPsP»CЏPµC, CЃCЂP°PIPSPёPIP°C,CЊ CЂP°P·PSC‹Pµ PjPsPґPµP»Pё CЂPµPiCЂPµCЃCЃPёPё PSP° PIP°C€PёC… PґP°PSPSC‹C….""")

# 1. P—P°PiCЂCґP·PєP° PґP°PSPSC‹C…
st.sidebar.header('1. P—P°PiCЂCґP·PєP° PґP°PSPSC‹C…')
uploaded_file = st.sidebar.file_uploader("P—P°PiCЂCґP·PёC,Pµ CSV C„P°P№P»", type=["csv"])

if uploaded_file is not None:
    data = pd.read_csv(uploaded_file)
    st.subheader('PµCЂPµPґPїCЂPsCЃPjPsC,CЂ PґP°PSPSC‹C…')
    st.write(data.head())

    # P’C‹P±PsCЂ PїCЂPёP·PSP°PєPsPI
    st.sidebar.header('2. P’C‹P±PsCЂ PїCЂPёP·PSP°PєPsPI')
    features = st.sidebar.multiselect('P’C‹P±PµCЂPёC,Pµ PSPµP·P°PIPёCЃPёPjC‹Pµ PїPµCЂPµPjPµPSPSC‹Pµ', data.columns)
    target = st.sidebar.selectbox('P’C‹P±PµCЂPёC,Pµ C†PµP»PµPICґCЂ PїPµCЂPµPjPµPSPSCґCЂ', data.columns)

    if features and target:
        X = data[features]
        y = data[target]

        # 2. P P°P·PґPµP»PµPSPёPµ PґP°PSPSC‹C…
        test_size = st.sidebar.slider('P P°P·PjPµCЂ C,PµCЃC,PsPIPsP№ PIC‹P±PsCЂPєPё (%)', 10, 40, 20)
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size/100, random_state=42)

        # 3. P’C‹P±PsCЂ PjPsPґPµP»Pё
        st.sidebar.header('3. P’C‹P±PsCЂ PjPsPґPµP»Pё')
        model_name = st.sidebar.selectbox('P’C‹P±PµCЂPёC,Pµ PjPsPґPµP»CЊ',
                                          ['P›PёPSPµP№PSP°CЏ CЂPµPiCЂPµCЃCЃPёCЏ',
                                           'P”PµCЂPµPIPs CЂPµC€PµPSPёP№',
                                           'PЎP»CґC‡P°P№PSC‹P№ P»PµCЃ'])

        # P”PёPїPµCЂPїP°CЂP°PjPµC,CЂC‹ PjPsPґPµP»PµP№
        if model_name == 'P”PµCЂPµPIPs CЂPµC€PµPSPёP№':
            max_depth = st.sidebar.slider('PњP°PєCЃPёPjP°P»CЊPSP°CЏ PiP»CґP±PёPSP°', 1, 20, 5)
            model = DecisionTreeRegressor(max_depth=max_depth)
        elif model_name == 'PЎP»CґC‡P°P№PSC‹P№ P»PµCЃ':
            n_estimators = st.sidebar.slider('PѓPsP»PёC‡PµCЃC,PIPs PґPµCЂPµPICЊPµPI', 10, 200, 100)
            max_depth = st.sidebar.slider('PњP°PєCЃPёPjP°P»CЊPSP°CЏ PiP»CґP±PёPSP°', 1, 20, 5)
            model = RandomForestRegressor(n_estimators=n_estimators, max_depth=max_depth)
        else:
            model = LinearRegression()

        # PЂP±CґC‡PµPSPёPµ PjPsPґPµP»PµP№
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

        # 4. P PµP·CґP»CЊC,P°C,C‹
        st.subheader('P PµP·CґP»CЊC,P°C,C‹ PjPsPґPµP»PёCЂPsPIP°PSPёCЏ')
        st.write(f'**PњPsPґPµP»CЊ:** {model_name}')

        # PњPµC,CЂPёPєPё
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)
        st.write(f'**MSE:** {mse:.2f}')
        st.write(f'**R2 Score:** {r2:.2f}')

        # P”CЂP°C„PёPє
        fig, ax = plt.subplots()
        ax.scatter(y_test, y_pred)
        ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)
        ax.set_xlabel('PCґC,PёPSPSC‹Pµ P·PSP°C‡PµPSPёCЏ')
        ax.set_ylabel('PµCЂPµPґCЃPєP·P·P°PSPSC‹Pµ P·PSP°C‡PµPSPёCЏ')
        ax.set_title('P“CЂP°C„PёPє PїCЂPµPґCЃPєP·P·P°PSPёP№')
        st.pyplot(fig)

        # P’P°P¶PSPsCґC,CЊ PїCЂPёP·PSP°PєPsPI (PґP»CЏ PґPµCЂPµPICЊPµPI)
        if hasattr(model, 'feature_importances_'):
            st.subheader('P’P°P¶PSPsCЃC,CЊ PїCЂPёP·PSP°PєPsPI')
            feat_importances = pd.Series(model.feature_importances_, index=features)
            fig2, ax2 = plt.subplots()
            feat_importances.plot.bar(ax=ax2)
            st.pyplot(fig2)
    else:
        st.warning('PµPsP¶P°P»CґP№CЃC,P°, PIC‹P±PµCЂPёC,Pµ PїCЂPёP·PSP°PєPё Pё C†PµP»PµPICґCЂ PїPµCЂPµPjPµPSPSCґCЂ')
else:
    st.info('PµPsP¶P°P»CґP№CЃC,P°, P·P°PiCЂCґP·PёC,Pµ CSV C„P°P№P» PґP»CЏ P°PSP°P»PёP·P°')
```