This is a classification problem. My goal is classifying users as adopted = True or False, using their features at signup.

I used the engagement table to find users who meet the adoption definition:

```
def is_adopted(group):
    logins = group['creation_time'].drop_duplicates().sort_values()
    for i in range(len(logins) - 2):
        if (logins.iloc[i + 2] - logins.iloc[i]).days <= 7:
            return True
    return False

# Filter and get unique user_ids
adopted_users = engagement.groupby('user_id').filter(is_adopted)['user_id'].unique()

# Add adoption label to users table
df_2['adopted'] = df_2['user_id'].isin(adopted_users)
```

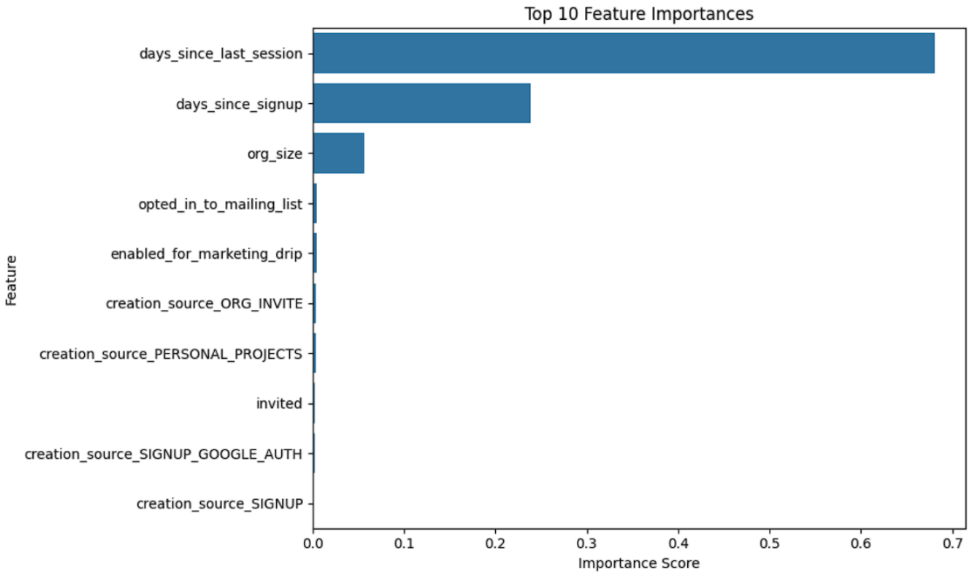
Next, I completed some feature engineering:

- **days_since_signup**: older accounts may be more likely to be adopted
- **days_since_last_session**: recent activity may indicate engagement
- **invited**: invited users might be more likely to use the product
- **org_size**: users in larger orgs might have different usage patterns
- **creation_source_***: how they signed up may influence adoption

I used a Random Forest Classifier:

- It naturally handles different types of data and doesn't require scaling
- Requires minimal preprocessing
- Gives strong baseline performance

I visualized the top 10 features to see what drives adoption the most.



Summary of Results:

- Users who signed up via ORG_INVITE have high adoption rates
- Users with large org_size are more likely to be adopted users
- Users invited by others tend to adopt more
- Time since last session is a good signal of churn vs adoption.