A.A. The alphabet is $\Sigma = \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{.}, \texttt{-}, \texttt{e}, \texttt{E}\}$.

A.B. My regular expression is

$$(\epsilon \cup -)DD^*(\epsilon \cup .DD^*)(\epsilon \cup (e \cup E)(\epsilon \cup -)DD^*).$$

The first chunk matches an optional `-`. The second chunk matches one or more digits — the integral part of the number. The next chunk matches an optional fractional part of the number. The final chunk matches an optional `e` or `E` construction.

B. No, A is not regular, as I now prove using the pumping lemma. Suppose for the sake of contradiction that A is regular. Let p be a pumping length. Let

$$w = a^p b^p c^{p^2}.$$

Then $w \in A$ and $|w| = p^2 + 2p \geq p$, so w can be pumped in the usual way. We know that $w = xyz$ where $|xy| \leq p$ and $|y| \geq 1$. Thus $y = a^k$ for some k such that $1 \leq k \leq p$. Then

$$xy^2z = a^{p+k}b^p c^{p^2} \notin A.$$

This contradiction implies that A is not regular.

C. Yes, A is regular, because it is the language of the regular expression $(01)^*$.

D. The PDA begins by non-deterministically guessing whether $i \neq j$ or $j \neq k$. That is, the start state has an ϵ-transition to a "top half" and an ϵ-transition to a "bottom half". In the top half, the PDA loads input as onto its stack, then matches input bs to stack as, then loops through any number of input cs, accepting if the matching worked (meaning $i \neq j$) and no errors (such as c before b) occurred. The bottom half is similar; the PDA loops through any number of as, loads bs onto its stack, then matches cs to those bs, accepting if the matching worked (meaning $j \neq k$) and no errors occurred.

E.A. FALSE. [Over $\Sigma = \{0, 1\}$, for example, $A = \Sigma^*$ is regular but $B = \{0^m 1^m : m \geq 0\}$ is not. Regularity is not about how "big" a language is. It's about how "complicated" a language is.]

E.B. TRUE. [We have proved so in class, or at least sketched the proof.]

E.C. TRUE. [We have proved so in homework.]

E.D. TRUE. [One way to see this is to think about the statement of the pumping lemma; if p is "good enough", then so is $p + 1$. Another way to see it is to think about the proof of the pumping lemma. The pumping length is the number of states in a DFA for A. But one can always add another state, that is disconnected from the rest of the state diagram. Such a state causes p to increment without changing A.]